AUTOMATIC OUTLIER DETECTION IN MULTIBEAM DATA

(Master thesis)

Natasha Hennis

September 2003

Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department Geodesy
Section Mathematical Geodesy and Positioning

Professor: Prof. Dr. Ir. P.J.G. Teunissen
Supervisor (TU Delft): Dr. R.C. Lindenbergh
Supervisor (Fugro Intersite B.V.): Ir. P. D. Bottelier

Preface

This master thesis is the result of my research performed at Fugro Intersite B.V., in the period from March till September 2003, for the final part of my study for Geodetic Engineer at the Delft University of Technology.

First I need to thank the people at Fugro Intersite B.V. for the opportunity to execute my research. Further I would like to thank my professor P.J.G. Teunissen, of the section Mathematical Geodesy and Positioning at Delft University of Technology, for his objective comments on my research. I especially want to thank my supervisors Peter Bottelier of Fugro Intersite B.V. and Roderik Lindenbergh of Delft University of Technology for always being available to give advice and help when needed. I must thank Henk van Buyten for seeking out adequate multibeam data sets and Kees de Jong for sharing his office with me and the other colleagues at Fugro Intersite B.V. for their help and amicability.
Not forgetting my friends I want to thank them for the much needed distraction and wonderful time. With great gratitude I thank my parents for their love, support, encouragement, help and everything else.

Leidschendam, September 2003

Natasha Hennis

Abstract

Multibeam echo sounding is a sea bottom survey method that produces abundant data. Since it is not essential to use all the acquired data for further processing data reduction algorithms have been developed. For these algorithms it is necessary that no outliers are present in the data. This demand is usually not met when working with multibeam data. Therefore outlier detection algorithms are needed.

The goal of this master thesis is to develop an algorithm that automatically removes outliers from a multibeam data set without removing the characteristics of the seabed or small objects that are located on the seabed like pipes.
After a literature study of existing procedures a 2D outlier detection algorithm has been developed. Although this 2D detection algorithm is primarily based on the 1D outlier detection algorithm developed by MSc. P.D. Bottelier various features of the existing procedures are incorporated in an altered form as well.

The 2D algorithm is designed to process data that is acquired by a calibrated multibeam system and corrected for systematic errors. Thus the data is already corrected for the attitude, tide and other systematic errors. The main feature of the 2D algorithm is that it uses cross validation with the aid of kriging. For kriging a covariance function, which is determined empirically from the measured depths of the data, is used.
The 2D algorithm starts with thresholding in order to detect gross blunders. The data is then read piece by piece into a buffer such that only a small portion of the whole data set is processed at once. The size of the buffer is variable and must be specified by the operator. For all the soundings present in the buffer a covariance function is established. This is done by first determining an empirical covariance function. It is then smoothed with a moving average operator so that an analytical covariance function can be established from it. After that the neighbours, that are to participate in the cross validation, are selected for each sounding. The calculation of the predicted depth is then performed with the equations for Ordinary Kriging with the selected neighbours. Finally the difference between the measured and predicted depth is tested against a test criterion.

In order to evaluate the performance of the 2D outlier detection algorithm roughly 70 tests have been executed using four different data sets. There was no ground truth available for the data sets used therefore the results are inspected visually and compared to the results of a reference method, the 1D outlier detection algorithm. As a first processing step thresholding has been applied to both algorithms prior to the execution of the tests. The reference method has the tendency to detect abundant false outliers which are mainly situated at the edges of a pipe. The tests performed were to determine the influence of the number of neighbours, the number of pings read into the buffer and the test criterion value.

Although it could be seen that the number of neighbours used in the 2D algorithm has an influence on the results it did not show a clear trend. However the influence of the number of beams read into the buffer is significant: the more pings that are read into the algorithm the less false outliers are detected. The test criterion is based on a confidence level and its value has a significant influence on the results. Based on visual inspection it can be said that: the smaller this confidence level, the smaller the number of falsely detected outliers but the number of undetected true outliers does increase, although not tremendously.
After comparison with the results of the reference algorithm the 2D algorithm usually detects at least 40% less false outliers if the test criterion is 1.96, for both algorithms, and if the 2D algorithm uses six neighbours and a buffer size of fifty pings.

Further research is recommended concerning the influence of the number of neighbours used and re-usage of the covariance function. The influence of the number and location of neighbours deserves further research since in the 1D algorithm a trend is present. Further research concerning the covariance function is in its place because it might be possible to calculate it for one buffer and re-use it for a number of consecutive buffers which could save computational time.

Samenvatting

Multibeam echoloding is een zee- of rivier bodem opnametechniek waarbij een grote hoeveelheid datapunten wordt gegenereerd. Omdat niet alle data punten noodzakelijk zijn voor verdere dataverwerkingsstappen zijn er data reductie procedures ontwikkeld. Zo'n reductie procedure gaat ervan uit dat er geen outliers in de data voorkomen. De meeste data verkregen met multibeam echoloding voldoet niet aan dit vereiste, dus is er eerst een methode nodig om de outliers te detecteren en te verwijderen.

Het doel van dit afstudeeronderzoek is het ontwikkelen van een algoritme waarbij outliers in multibeam echolood data gedetecteerd en verwijderd worden zonder dat karakteristieke kenmerken van de zeebodem of kleine objecten, zoals pijpleidingen op de zeebodem, verloren gaan.
Na een literatuur studie van bestaande procedures is er een 2D algoritme ontwikkeld. Dit algoritme is hoofdzakelijk gebaseerd op het 1D algoritme ontwikkeld door Ir. P.D. Bottelier en is aangevuld met verschillende kenmerken van andere procedures.

Het 2D algoritme is ontwikkeld voor data afkomstig van een gecalibreerd multibeam systeem. De data moet dus al gecorrigeerd zijn voor de bewegingen van het schip. Ook moet er al voor getijdebewegingen en andere systematische foutenbronnen gecorrigeerd zijn. Het belangrijkste kenmerk van het 2D algoritme is dat cross-validatie met behulp van kriging gebruikt wordt. Voor kriging wordt een covariantie functie gebruikt, die gebaseerd is op de gemeten diepten.
De eerste stap van het 2D algoritme is het detecteren van blunders met behulp van diepte drempel waardes. Vervolgens wordt de data set in delen verwerkt in zogenaamde buffers. De buffer grootte is door de gebruiker in te stellen. Voor elke buffer wordt er vervolgens een covariantie functie bepaald. Dit wordt gedaan door eerst een empirische covariantie functie vast te stellen die vervolgens gladder wordt gemaakt met behulp van een voortschrijdend gemiddelde. Aan de hand van deze gladdere empirische covariantie functie wordt een analytische covariantie functie bepaald. Vervolgens wordt voor elke data punt de buurpunten bepaald die meegenomen worden bij het krigen. De berekening van de voorspelde diepte wordt uitgevoerd met behulp van de formules voor Ordinary Kriging. Hierbij worden alleen de geselecteerde buurpunten gebruikt. Als laatste stap wordt het verschil tussen de gemeten en voorspelde diepte vergeleken met een test kriterium.

Om de prestatie van het 2D outlier detectie algoritme te bepalen zijn ongeveer 70 tests uitgevoerd op vier verschillende data sets. Aangezien er geen 'ground truth' was zijn de resultaten visueel geïnspecteerd en vergeleken met de resultaten van een referentie algoritme, het 1D outlier detectie algoritme. Voor beide algoritmes zijn van te voren extreme blunders verwijderd met dezelfde drempelwaardes. Het referentiealgoritme heeft de neiging om verkeerde outliers te detecteren, met name bij de randen van een pijpleiding. De uitgevoerde tests hadden voornamelijk tot doel om de invloed van het aantal buurpunten, het aantal profielen in een buffer en de waarde voor het test kriterium te bepalen.

De resultaten betreffende het aantal buurpunten gebruikt bij het 2D algoritme wijst uit, dat het verschil uitmaakt hoeveel buurpunten gebruikt worden, hoewel geen duidelijke trend aanwezig is. Het aantal profielen in een buffer beïnvloedt de resultaten zeer zeker. Hierbij is een trend waargenomen: hoe meer profielen gebruikt worden in de buffer des te minder outliers ontdekt worden. De waarde voor het test kriterium wordt gebaseerd op een betrouwbaarheidsinterval en heeft een significante invloed op de resultaten. Na visuele inspectie kan gezegd worden dat hoe kleiner het betrouwbaarheidsinterval des te kleiner is het aantal verkeerd gedetecteerde outliers. Het aantal niet gedetecteerde outliers wordt dan wel groter alhoewel niet significant veel groter.

Na vergelijking van de resultaten van het 1D en 2D algoritme is te zien dat het 2D algoritme voor de eerste drie datasets 40% minder verkeerde outliers detecteert wanneer voor beide algoritmes een test kriterium van 1.96 en voor het 2D algoritme 6 buren en 50 profielen per buffer gebruikt worden.

Verder onderzoek wordt aanbevolen betreffende de invloed van het aantal buren en de lokatie van de buurpunten. Wellicht is het dan mogelijk een duidelijke trend op te sporen, zoals er een bestaat voor het 1D algoritme.
Ook zou het interessant zijn uit te zoeken of het mogelijk is om, zonder kwaliteitsverlies, de covariantie functie te berekenen voor één buffer en vervolgens te gebruiken voor een aantal opeenvolgende buffers. Dit zou de benodigde computer berekeningstijd kunnen verminderen.

Contents

1 Introduction

In marine geodesy multibeam echo sounding is an ever more used and improved survey technique to determine the bathymetry of the seabed. It is a technique that allows measuring the seabed swiftly and in great detail. The amount of data acquired is numerous and often more than fundamentally needed to represent the surveyed area. This is especially the case where there is barely any height variation in the bathymetry, e.g. a fairly flat seabed. If all the acquired data is used to make, for instance, a Digital Terrain Model (DTM), it will take a lot of DTM processing time. In order to minimize the processing time, it is preferred to have a representative data set instead of the complete data set. This preference calls for data reduction algorithms. To ensure that these algorithms perform efficiently and accurately the data set must be devoid of erroneous measurements.
Multibeam data usually has some erroneous measurements due to reflections upon fish, air bubbles and suspended debris. This means that prior to data reduction it is necessary to check the data on outliers and remove them accordingly. To do this manually will take a great deal of time and consequently from an economical point of view an automatic outlier detection procedure is a must.

The goal of this master thesis is to develop an algorithm that automatically removes outliers from a multibeam data set without removing the characteristics of the seabed or small objects that are located on the seabed like pipes. It should be applicable for multibeam data acquired by any type of multibeam system.
The algorithm is developed for post-processing but during the development the constraint that it should be real-time or near real-time applicable is kept in mind. The algorithm is developed in such a way that it can be converted into a real-time algorithm without violating the original procedure too much. The algorithm is programmed and extensively tested in MATLAB software. An optimal implementation of the procedure with respect to CPU time is beyond the scope of this thesis and will thus not be investigated.
The developed algorithm is based mainly on the procedure developed by MSc. P.D. Bottelier during his master thesis concerning data reduction procedures, [Bottelier*, 1998*]. Yet, in order to develop an algorithm that meets the above mentioned constraints an inventory of current outlier detection methods (automatic and semi-automatic) has been made based on a literature study. Some useful parts of these procedures have been applied in a modified form in the developed algorithm. The algorithm was developed and tested using diverse multibeam data sets; data sets from different multibeam systems and from different locations, with and without structures such as pipes.

This master thesis has the following structure. The principles of multibeam echo sounding are described in Chapter 2. A brief explanation about cross-validation and kriging follows in Chapter 3. An overview of current outlier detection procedures is given in Chapter 4 and the developed outlier detection procedure in Chapter 5. The tests and results will be treated in Chapter 6 and the conclusions and recommendations will be given in Chapter 7.

2 Multibeam echo sounding

Multibeam echo sounding is a survey technique that provides a rapid means of determining the bathymetry of the seabed, a riverbed or lake bottom. The first multibeam echo sounding system became operational in 1982 on an academic research ship. It is the successor of singlebeam echo sounding. An overview of the multibeam echo sounding theory will be given in this chapter, starting with the fundamentals of echo sounding in Section 1. The concept of the multibeam echo sounding system will be discussed in Section 2 and in the last section, Section 3, errors related to multibeam echo sounding systems will be discussed briefly. The main source of information for this chapter is [De Jong et al, 2002].

2.1 Echo sounding

Echo sounding is based on the principle that water is an excellent medium for the transmission of sound waves and that a sound pulse will reflect from the bottom of a water mass to all directions and thus return to its source as an echo. The depth of the water mass can then be determined from the time interval between the initiation of the sound pulse and the reception of the echo.

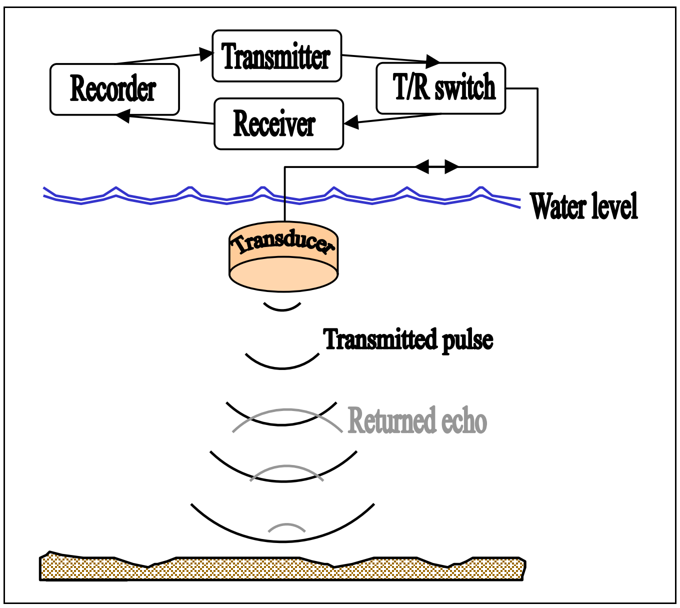

Figure 2.1 Echo sounding concept

An echo sounding device usually consists of a transmitter, a transmitter/receiver (T/R) switch, a transducer, a receiver and a recorder. The transmitter generates a sound pulse and the T/R switch passes the power from the transmitter to the transducer and consequently specifies the length of the pulse. The transducer, which is usually mounted on the hull of a ship, converts the electrical power into acoustical power and sends the acoustic signal into the water. It also receives the echo and converts it back into an electrical signal. The echo signal is then amplified by a receiver and sent through to the recording system. The recorder controls the signal emission, measures the two-way travel time of the acoustic signal, stores the data and converts the time interval into a range using the measured speed of sound in the water column, see Equation 2.1. After receiving the echo a new pulse is generated and the cycle is repeated. One such a cycle is called a 'ping'. A schematization of this cycle is given in Figure 2.1.

$$r = \tfrac{1}{2}ct \qquad \textbf{Equation 2.1}$$

in which:

r = range or distance
t = two-way travel time
c = sound velocity in water

2.2 The multibeam concept

Multibeam echo sounder systems (MBES) are divided into two groups, *swath* systems and *sweep* systems. A swath system essentially produces multiple acoustic beams from a single transducer system whereas a sweep system mainly consists of an array of singlebeam echo sounders mounted on booms, which are arranged perpendicularly to the sides of a vessel. A swath system can be mounted on the hull of a vessel or on the hull of a remotely operated vehicle (ROV). It is the most common MBES system. This section will thus mainly go into the concept of a swath system and fleetingly into the concept of a sweep system.

Swath system

A swath MBES system transmits an acoustic pulse that resembles a fan. It is wide in one direction (across track) and narrow in the perpendicular direction (along track). The echo is received by a transducer, which segments it into multiple smaller beams, as shown in Figure 2.2. The widths of these beams are in the order of one to a few degrees, depending on the system.

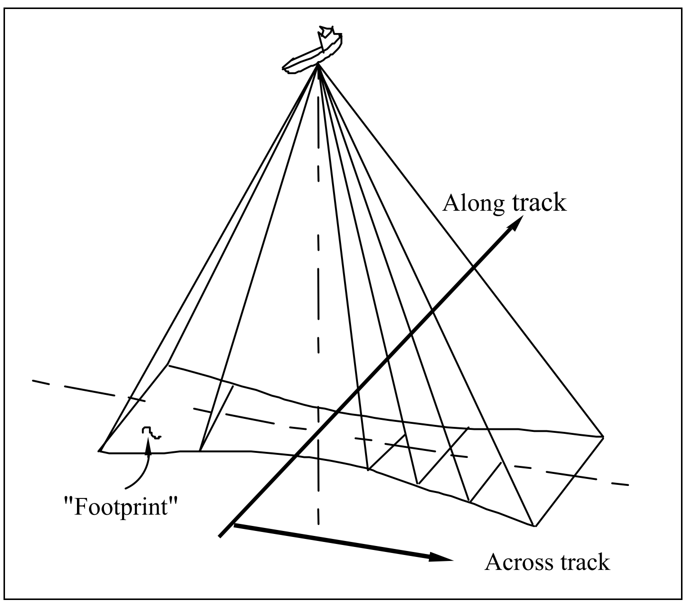

Figure 2.2 MBES footprints

Per pulse transmission (ping) a high number of depths are thus generated. From a single vessel's track, a band of depths are obtained opposed to a single line of depths obtained from a singlebeam echo sounder. The coverage (footprints) obtained by a multibeam system and by a singlebeam system is illustrated in Figure 2.3.

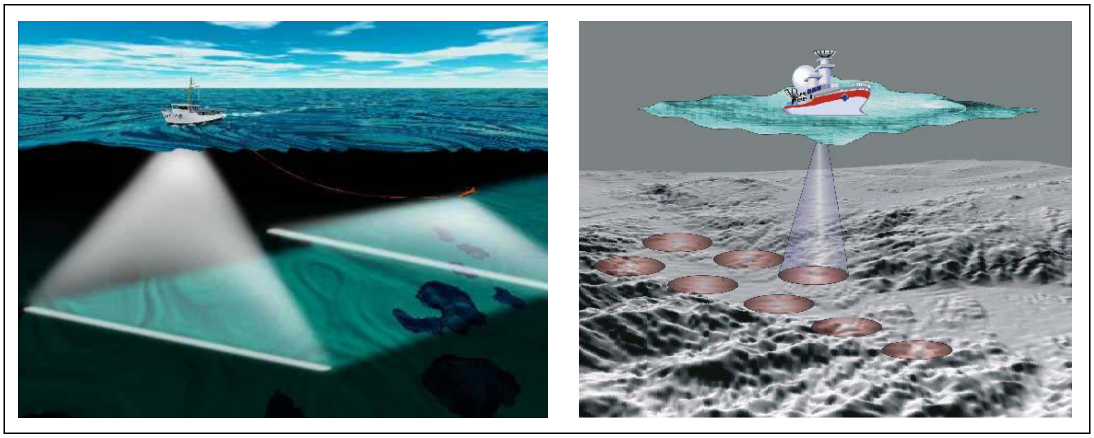

Figure 2.3 Multibeam coverage (left) and Singlebeam coverage (right)

For each beam received there is a two-way travel time (t) and a beam angle (ψ). Neglecting errors and vessel motion these measurements can be converted into depths (D) and across-track positions (y) of the soundings using:

$$D = \tfrac{1}{2} ct \cos\psi \qquad \textbf{Equation 2.2}$$

$$y = \tfrac{1}{2} ct \sin\psi \qquad \textbf{Equation 2.3}$$

After correcting for the ships movement and the systematic errors each sounding has three (x, y and z) coordinates referenced to the position of the transducer. These coordinates are then transformed to local x, y and z coordinates using the position of the transducer (vessel) at the moment of the measurement.

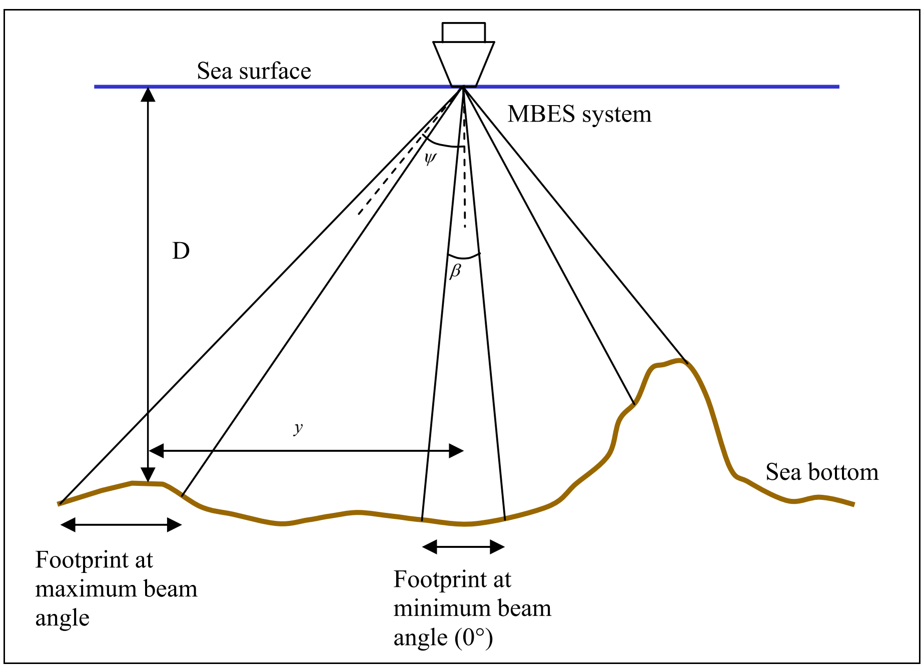

Figure 2.4 MBES footprint size versus beam angle ψ

The footprint of an MBES system increases with the beam angle. For the vertical sounding the beam angle (ψ) is 0° and the footprint is the smallest, see Figure 2.4. The footprint also depends on the beam width (β) and the depth of the water column. The bigger either of these parameters, the bigger the footprint becomes. The size of a detectable feature depends on the size of the footprint, the angle of incidence and the slope of the seabed.

Sweep system

A sweep MBES system consists of a number of singlebeam echo sounders fitted equidistantly on booms, which are fixed perpendicular to the motion of the survey vessel (Figure 2.5). Each singlebeam produces one beam with constant width and is directed straight down. The number of beams thus depends on the number of singlebeam echo sounders used. The percentage of the bottom covered depends on the spacing of the single beams, the beam widths and the water depth. A 100% coverage can be obtained if the spacing and beam width is adapted to the water depth.

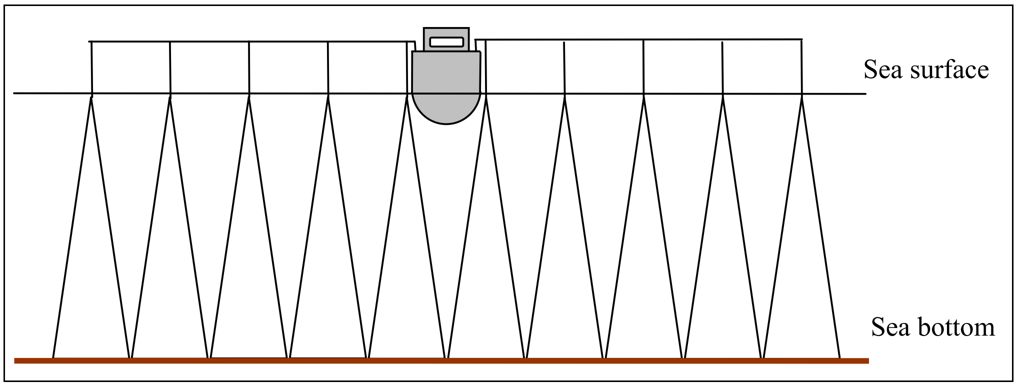

Figure 2.5 A sweep system

2.3 Multibeam errors

The total list of MBES errors and error sources is extensive and therefore only the most important errors will be discussed.

The vessel's pitch, roll, heave and heading (yaw) are important error sources.
Pitch is the rotation around the y-axis, the vessel's across-track axis. An area will be illuminated at a position behind or before the expected survey line, see Figure 2.6a. It causes errors in the x and z coordinates of the soundings as shown in Figure 2.6b.

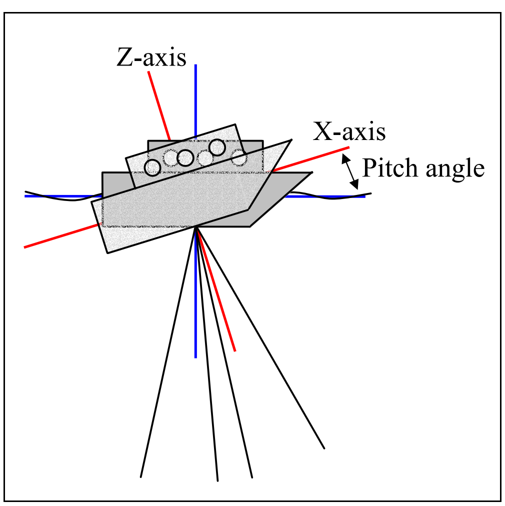

Figure 2.6a Pitch

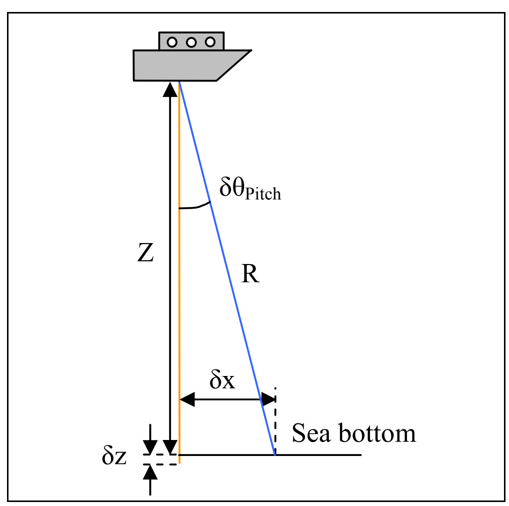

Figure 2.6b Geometry of the pitch errors

Roll is the rotation around the x-axis, the vessel's along-track axis. The beam angle will differ from what is expected and thus the area illuminated, see Figure 2.7a. It causes the y and z coordinates of the soundings to be erroneous as shown in Figure 2.7b.

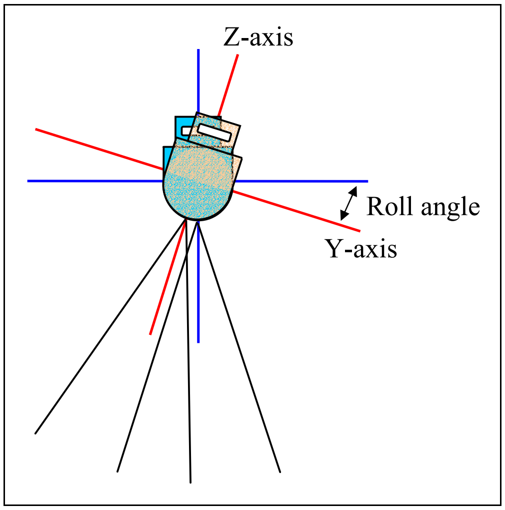

Figure 2.7a Roll

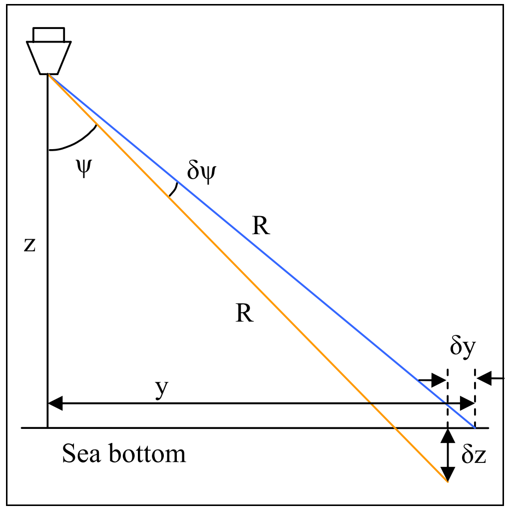

Figure 2.7b Geometry of the roll errors

Heave is the vertical motion of the vessel. As reference an arbitrary plane is used, e.g. the idealized flat sea surface. The distance from the bottom to the vessel will be greater or smaller and the illuminated area will consequently be greater or smaller, see Figure 2.8a. It causes biases in the z and x coordinates of the soundings as shown in Figure 2.8b.

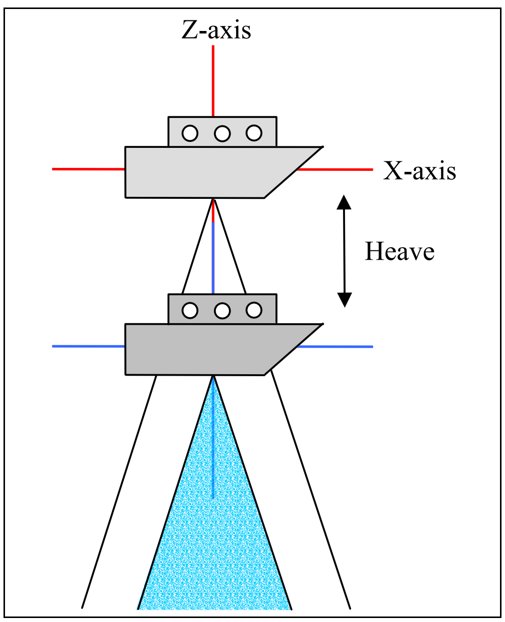

Figure 2.8a Heave

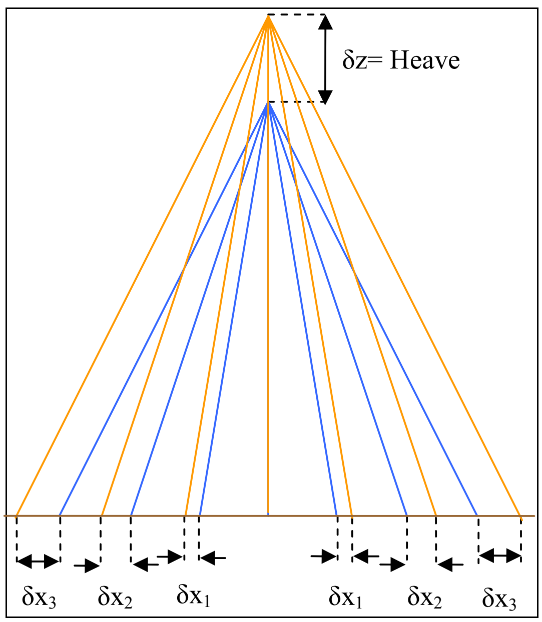

Figure 2.8b Geometry of the heave errors

Heading is the rotation around the z-axis. It causes the alignment of the illuminated areas to differ from what is expected and thus the x en y coordinates are biased, see Figures 2.9a and 2.9b.

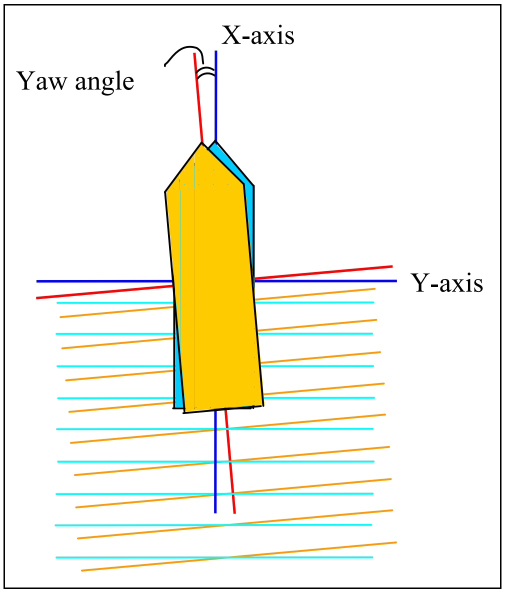

Figure 2.9a Heading

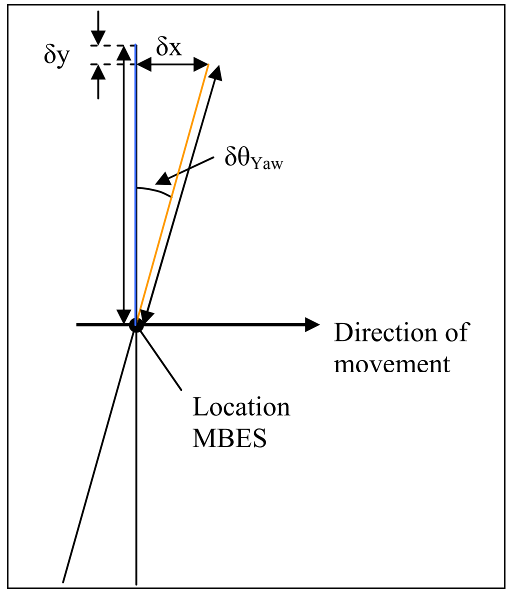

Figure 2.9b Geometry of the heading errors

Motion sensors measure the roll, pitch, heave and heading in order to estimate the errors and then correct for them.
The systems electronics causes measurement errors and beam angle errors which results in interval and beam angle biases. By performing a calibration of the system, before a survey starts, estimations of these errors are obtained in order to correct for them.
The sound velocity is also an error source. If the speed of sound at the transducer is measured incorrectly then the beam angles are different then expected. Another error related to the sound velocity is that if the speed of sound in the water column is not represented correctly then the depth measurements are biased. Both errors occur due to acoustic propagation characteristics. In order to minimize these errors it is customary to constantly measure the speed of sound at the transducer and to measure a profile of the speed of sound before each survey and if necessary during the survey, depending on the duration of the survey.

3 Cross validation & Kriging

Kriging is a well known interpolation method and one of its applications is in cross validation. In order to apply kriging we use a covariance function. These three topics are thus related to each other and will briefly be enlightened in this chapter. Cross validation will be discussed in Section 1 and kriging in Section 2. In the final section, Section 3, the covariance function will be treated. Sections 2 and 3 are primarily based on [Isaaks et al, 1989] and Section 3 is also based on [De Min, 1996].

3.1 Cross validation

Cross validation refers to the comparison of the value of a measured data point and the predicted value at the same location. The predicted value is obtained by using the measured values of data points in a certain neighbourhood; see for an example Figure 3.1. The difference between the two values gives an indication of the reliability of the measured value of the cross validated point. If the difference is greater than a certain value, the test criterion, then that point is considered an outlier. In multibeam data an outlier can be defined as a data point, also called ‘sounding’, which Z-coordinate deviates from those of its direct neighbours in such a way that it disturbs the image of the bottom topography [Bottelier, 1998].

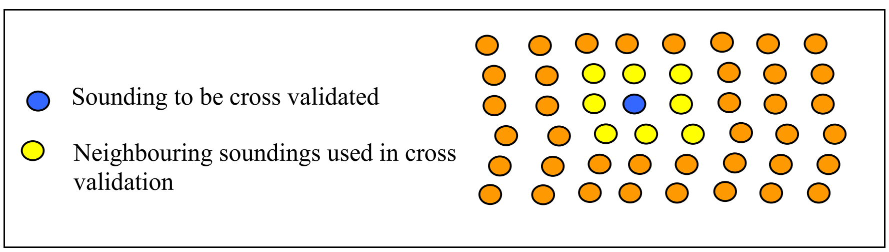

Figure 3.1 Neighbours for cross validation (Top view of 6 pings with 8 beams each)

Interpolation

Interpolation is a method to estimate the value of a point out of the values of neighbouring data points. There are different interpolation methods with each different characteristics. In the following, three methods will be given with a short explanation of how they work.

- Nearest neighbour
  In this method the predicted value is obtained from one sounding, the sounding with the minimum horizontal distance to the sounding to be predicted. The depth of this nearest neighbour is then the predicted depth. Although this method is rapid, it does not give a realistic predicted value.
- Inverse distance
  In this method a number of neighbouring soundings are used to obtain a predicted value. Each neighbour contributes to the predicted depth according to a weight, which is proportional to the inverse of the horizontal distance between the neighbour and the prediction sounding. This method requires more time then the previous method, but it gives a more realistic predicted value since it takes the bottom characteristics, as expressed by the neighbouring soundings, into account. It still does not give the most reliable interpolation result for certain applications.

- Kriging
  In this method the predicted value is again obtained from a number of neighbouring soundings and each sounding contributes to the predicted depth according to a weight. The difference to the previous method is that the weights are obtained from an analytical covariance function. The weights are dependent on the horizontal distance between the neighbour and the prediction sounding, and the covariance function. A covariance function gives the correlation between two soundings (see Section 4). The obtained prediction value is realistic since it takes the bottom characteristics into account, which is described by the covariance function. Its disadvantage is that it requires more computing time than the previous methods.

Computation of predicted values

Before the computation of the predicted values the mean depth is subtracted from the measured depth values resulting in depth differences, $dZ_i = Z_i - \overline{Z}$.

In general a predicted value is obtained by summation of the weights (ω) of the neighbouring soundings multiplied by the depths of the soundings:

$$d\hat{Z}_p = \sum_{i=1}^{Q} \omega_{Pi} \cdot dZ_i \qquad \textbf{Equation 3.1}$$

in which:

$d\hat{Z}_p$ = predicted depth difference of sounding P
P = interpolation sounding
i = neighbouring sounding
Q = number of neighbouring soundings
ω_{Pi} = weight of neighbouring sounding for interpolation sounding P
dZ_i = measured depth of neighbouring sounding *i* after subtraction of the mean depth

Since for each predicted depth the measured depth is known as well, it is possible to compute, for each sounding, the interpolation or prediction error (ε_p), the difference between the measured and predicted depth:

$$\varepsilon_P = dZ_P - d\hat{Z}_P = dZ_P - \sum_{i=1}^{Q} \omega_{Pi} \cdot dZ_i \qquad \textbf{Equation 3.2}$$

3.2 Kriging

From the interpolation methods mentioned kriging is the only method that is a best linear unbiased estimator, B.L.U.E [Isaaks et al, 1989]. The fact that the estimates obtained from kriging are linear combinations of the available data makes it ‘linear’. It is ‘unbiased’ since it expects to have a mean residual error equal to zero. Its aim of minimizing the variance of the prediction errors makes it ‘best’. This last factor of minimizing the variance of the prediction error is the factor that distinguishes it from most other methods.

[Isaaks et al, 1989] prove, in their Chapter 12, that by using Equation 3.3 in combination with the constraint that the sum of the weights must equal one, Equation 3.4, that the variance of the prediction error is minimized.

$$\sum_{j=1}^{Q} \omega_{Pj} C_{ij} + \mu = C_{Pi} \qquad \text{for} \quad i = 1 \ldots \text{Q} \qquad \textbf{Equation 3.3}$$

$$\sum_{j=1}^{Q} \omega_{Pj} = 1 \qquad \textbf{Equation 3.4}$$

in which:

$$\left.\begin{aligned} C_{Pi} &\equiv C(Pi) \\ C_{ij} &\equiv C(s_{ij}) \end{aligned}\right\} \text{Obtained from a covariance function (Chapter 4)}$$

In following equation the above set of equations (3.3 and 3.4) is written in matrix notation:

$$\begin{pmatrix} C_{11} & C_{12} & \Lambda & C_{1Q} & 1 \\ C_{21} & C_{22} & \Lambda & C_{2Q} & 1 \\ M & M & O & M & M \\ C_{Q1} & C_Q 2 & \Lambda & C_{QQ} & 1 \\ 1 & 1 & \Lambda & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \omega_{P1} \\ \omega_{P2} \\ M \\ \omega_{PQ} \\ \mu \end{pmatrix} = \begin{pmatrix} C_{P1} \\ C_{P2} \\ M \\ C_{PQ} \\ 1 \end{pmatrix} \qquad \textbf{Equation 3.5}$$

$$\mathbf{C} \quad \cdot \quad \omega \quad = \quad \mathbf{D}$$

The Lagrange multiplier (μ) can be interpreted as a scale factor applied to the weights (ω) as a direct consequence of the constraint that the sum of the weights must equal one. It can be seen that the weights are only a function of the values of the covariance function C(s) since $\omega = C^{-1} \cdot D$ and the matrices C and D are derived from covariance values only. The variance of the prediction error can then be calculated according to the following equation.

$$\sigma^2{}_{prediction} = C(0) - D^T \alpha = C(0) - D^T C^{-1} D \qquad \textbf{Equation 3.6}$$

The above-described method of kriging is known as Ordinary Kriging and is applicable to data if the expected value for each data point is not known and if the data does not express a trend. Simple Kriging and Universal Kriging are two other well-known kriging methods. Simple Kriging is used if the expected value for each data point is known and if there is no trend present. Universal Kriging is applicable to data in which the expected value for each data point is not known and the data may express a trend, [Chilès et al, 1999].

Ordinary Kriging is in fact nearly identical to least squares prediction. Least squares prediction assumes, just like Simple Kriging, that the expected value for each data point is known.

Least squares prediction: $\displaystyle\sum_{j=1}^{Q} \omega_{Pi} C_{ij} = C_{Pi}$ for i=1…Q **Equation 3.7**

In matrix notation:

$$\begin{pmatrix} C_{11} & C_{12} & \Lambda & C_{1Q} \\ C_{21} & C_{22} & \Lambda & C_{2Q} \\ M & M & O & M \\ C_{Q1} & C_Q 2 & \Lambda & C_{QQ} \end{pmatrix} \cdot \begin{pmatrix} \omega_{P1} \\ \omega_{P2} \\ M \\ \omega_{PQ} \end{pmatrix} = \begin{pmatrix} C_{P1} \\ C_{P2} \\ M \\ C_{PQ} \end{pmatrix} \qquad \textbf{Equation 3.8}$$

$$\boldsymbol{C_{ij}} \quad \cdot \quad \boldsymbol{\alpha_{Pi}} \quad = \quad \boldsymbol{C_{Pi}}$$

Looking at Equation 3.7 (least squares prediction) and Equations 3.3 and 3.4 (Ordinary Kriging) it is possible to see that the only difference is that Ordinary Kriging has an additional constraint on the weights. Least squares prediction does not have the constraint that the sum of the weights must equal one, [Moritz, 1980].

3.3 Covariance function

A covariance function describes spatial variability. In relation to this subject it describes the spatial variability of the depth, thus it gives the correlation of the depths between two data points (soundings) separated from each other with a distance *s*. In the following a description will be given on how to obtain a covariance function from a data set.

In order to determine an analytical covariance function from a data set, an empirical covariance function has to be determined first. Prior to determining an empirical covariance function the mean signal value must be subtracted from the data. The mean signal value is usually the average of all the measured data points. In relation to multibeam data, this means that the average of all the depths should be subtracted from every sounding according to:

$$dZ_i = Z_i - \overline{Z} \qquad \text{for i =1…N} \qquad \textbf{Equation 3.9}$$

in which:

N = number of soundings in the data set

Z_i = depth of sounding *i*

$\overline{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$, the average depth of all the soundings

If all the data is given on an equidistant grid and over the whole domain then the covariance function is calculated according to:

$$C(s) = \frac{1}{N}\sum_{i=1}^{N} dZ_i dZ_{i'} \qquad \textbf{Equation 3.10}$$

in which:

- N is the number of products between two soundings used to compute the covariance *C* at separation distance *s*. This number is equal to the number of pairs of soundings at distance *s*.
- Sounding *i*’ and *i* form a pair if the distance between them is equal to *s*

Multibeam data cannot be considered as data given on an equidistant grid because the footprint of a beam increases with the beam angle and thus the distance between two consecutive soundings increases with the beam angle as well. Thus it is not possible to use Equation 4.2 and an alternative equation is needed. The computation must be carried out with the usage of distance intervals *ds* with discrete s values at the middle of the interval. If the distance between two soundings lies between $s - \frac{1}{2}ds$ and $s + \frac{1}{2}ds$ it is used to calculate a value for the covariance function at distance *s*. See for a graphical clarification Figure 3.2. The number of pairs of soundings, n_s, used to compute the covariance at distance *s* depends on the distance *s* and on the width of the interval. The absolute covariance values derived from multibeam data can thus be calculated empirically by the following equation.

$$C_{1,n_s}(s) = \frac{1}{n_s}\sum_{i=1}^{n_s} dZ_i dZ_{i'} \qquad \textbf{Equation 3.11}$$

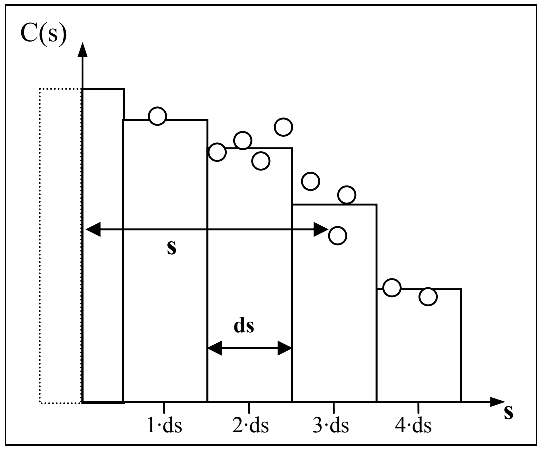

Figure 3.2 Calculation of empirical covariance values using distance intervals

If all the soundings do not participate, or not as often as others, in the calculation of the covariance function at distance *s*, the equation above (Equation 3.11) is not the best one to use, [De Min, 1996]. The reason is that especially for short distances, where the correlation is the strongest, it is important that the covariance function describes the magnitude that the function value can change over a certain distance and not what the absolute value of the covariance is. It must describe the percentage of correlation for two soundings at a certain distance from each other. This correlation percentage must be related to the variance of the complete data set, the covariance at distance zero (C(0)), which approximates the variance (σ^2). Thus for the calculation of the empirical covariance function for multibeam data yet a different equation is needed thus calling for:

$$C_{n_s}(s) = \frac{C_{1,n_s}(s)}{C_{1.n_s}(0)} C(0) \qquad \textbf{Equation 3.12}$$

with:

$$C_{1,n_s}(0) = \frac{1}{2n_s} \sum_{i=1}^{n_s} (dZ_i^{\ 2} + dZ_{i'}^{\ 2})$$

and

$$C(0) = \frac{1}{N} \sum_{i=1}^{N} dZ_i^{\ 2} \approx \sigma^2$$

The covariance function can have the tendency to be irregular due to point noise, which is the random deviation that is present in each depth measurement [Bottelier, 1998]. This will especially be the case if the number of soundings used to compute the covariance function empirically is relatively small. The number of soundings used to determine the covariance function depends on the type of system used to acquire the data and the size of the area for which the covariance function is calculated. If the number of soundings used is only 60 the covariance function will tend to be more erratic then if e.g. 300 soundings was used.
To decrease the irregular tendency a moving average filter of five points is applied to the covariance function according to the following equation.

$$\widetilde{C}_i = \frac{C_{i-2} + C_{i-1} + C_i + C_{i+1} + C_{i+2}}{5} \qquad \textbf{Equation 3.13}$$

Using Equation 3.13 the first two and the last two covariance values are not smoothed. It is not necessary to smooth them either since the first two covariance values are used to estimate the point-noise and the last two covariance values are ignored because they represent the correlation of soundings at relatively large distances which are small and usually not considered.

In the interpolation method Kriging, the weights attached to the neighbouring soundings are calculated with the aid of an analytical covariance function as discussed in the previous section. Kriging needs the inverse of the covariance matrix, which existence is guaranteed if the analytical covariance function is positive-definite. The function obtained from Equation 3.14 [Bottelier, 1998] has the required properties and is therefore used for the approximation of the analytical covariance function. Numerous tests show that for multibeam data it represents the empirical covariance function better in the first part than towards the end. That it does not represent the empirical covariance function as good in the end as in the beginning is not of great concern since in the cross validation only close neighbours will be used. This implies that only the first part of the covariance function will be used.

$$C(s_{ij}) = C(0) \cdot (1 - f) \cdot e^{-f} \qquad \textbf{Equation 3.14}$$

with:

$$f = \left(\frac{s_{ij}}{d} \right)^{\kappa}$$

and

$$\kappa = \frac{\log(0.3149)}{\log\left(\frac{\xi}{d} \right)}$$

in which:

s_{ij} = horizontal distance between two soundings
d = zero crossing of the covariance function
κ = curvature at C(0)
ξ = correlation length
C(0) = variance belonging to distance zero

In order to acquire the analytical covariance function two parameters have to be computed from the empirical covariance function, the zero passage (d) and the correlation length (ξ). These parameters are calculated using the smoothed covariance values. The zero crossing is the distance where the covariance value is equal to zero for the first time; C(d) = 0. The correlation length is the distance where the covariance is, for the first time, half the value of the covariance at distance zero; C(ξ) = 0.5*C(0).

The covariance function acquired according to the above-mentioned approach is stochastic. But in order to cope with the propagation of the variances in further calculations the covariance function is assumed deterministic.
The covariance is not the only technique that can be used to describe the spatial continuity. The variogram is another and can be seen as a measure for the similarity of the terrain between two points. It can be calculated empirically according to the next equation.

$$\gamma(s) = \frac{1}{2n_s} \sum_{i=1}^{n_s} (dZ_i + dZ_{i'})^2 , \qquad \textbf{Equation 3.15}$$

Though the covariance function and the variogram are strictly two different measures of the spatial variability there is a relationship between the function values of the covariance function and the variogram function. The covariance function can be converted into a variogram according to the following equation.

$$\gamma(s) = C(0) - C(s)\,, \qquad \textbf{Equation 3.16}$$

$$V(s) = \gamma(\infty) - \gamma(s)\,, \qquad \textbf{Equation 3.17}$$

The reverse (Equation 3.17), thus calculating the covariance from a variogram is not always possible since the variogram does not always reach a maximum value, [Isaaks et al, 1989].

4 Current outlier detection procedures

The detection of outliers used to be done manually, but since the amount of data to be processed increased dramatically the time needed to detect the outliers increased as well. To reduce the time needed for the outlier detection, procedures have been invented to detect outliers automatically or semi-automatically. In this chapter an overview of several of these procedures will be given. In each of the first five sections a procedure will be described based on an article or thesis published by the developers. Then in the last section, Section 4.7 a short discussion about these procedures will be given. The results obtained by the procedure described in Section 1 will be used to compare the results of the 2D outlier detection algorithm (the newly developed algorithm).

4.1 1D outlier detection

This procedure is developed by P.D. Bottelier for his master thesis, [Bottelier, 1998]. It is divided into two consecutive steps and applied on data corrected for the navigation and motion of the survey vessel. If a sounding does not meet the criterion in any of the two steps it is recognized and tagged as an outlier.

Process:
The first step is used for extreme blunder detection. Blunders are soundings which are situated at more than two times the standard deviation of the depths referenced to the mean depth. The technique used in this procedure to remove blunders is thresholding. This implies that all depths that are deeper than $Z_{\max}$ or shallower than $Z_{\min}$ are removed. $Z_{\max}$ and $Z_{\min}$ are a priori determined values.

In the second step, the outlier detection step, soundings which Z coordinate deviates from those of its direct neighbours in such a way that it disturbs the image of the bottom topography is removed. The technique used for this step is cross-validation and the required predicted values are obtained using kriging. The difference between the measured and predicted values is a measurement for the reliability of the sounding being evaluated as mentioned in Chapter 3. If this difference exceeds a test criterion then the sounding is considered an outlier and removed from the data set. The following equation gives the test used to determine if the sounding is an outlier or not.

$$\frac{\left|Z_{measured,i} - Z_{predicted,i}\right|}{\sqrt{\sigma^2_{noise} + \sigma^2_{prediction}}} > C_1 \qquad \textbf{Equation 4.1}$$

In this procedure the test criterion that is used is based on the assumption that the differences are normally distributed. The test criterion is also based on a 5% confidence level which yields a critical value of 1.96. A 5% confidence level implies that 5% of the data may be detected as outliers. The cross validation and consequently the interpolation is done per ping and for kriging four neighbouring points are used which are equally distributed on both sides of the interpolation point.
Per ping a covariance function is determined empirically in order to determine the random deviation present in each depth measurement (σ^2_{noise}) and the values of the weights attached to a

number of neighbouring soundings used in the interpolation method. The interpolation method in conjunction with the covariance function yields a value for the variance of the prediction error ($\sigma^2_{prediction}$).

4.2 Outlier detection using neighbourhood average

This procedure is developed by the National Oceanic and Atmospheric Administration (NOAA) and is executed immediately upon reading data from the raw file, thus before corrections are applied and before initiation of a sounding selection process, [Herlihy et al, 1992]. Whether to determine a sounding as erroneous or not is based on three main grounds.

Grounds to determine if a sounding is erroneous:

1. The depth or cross track distance is identified as a gross blunder.
2. The difference between the depth and neighbourhood average exceeds the specified limit.
3. The sounding does not have enough non-zero neighbours to compute a meaningful neighbourhood average.

Process:
In this process the sounding to be evaluated is examined relative to the weighted average of all non-zero depths in the 'neighbourhood' of the evaluated sounding. The NOAA has specified the neighbourhood for two types of multibeam data, multibeam data from the Sea Beam and multibeam data from the Hydrochart II. The neighbourhood for the Sea Beam is built up from the current ping, the ping containing the evaluated sounding, and the three pings immediately before and after the current ping. For the Hydrochart II the neighbourhood is built up from the current ping and the six pings immediately before and after the current ping. For both types of data the neighbourhood is further built up from soundings from the current beam; the beam corresponding to the beam number of the evaluated sounding and the two adjacent beams. This results in a neighbourhood of 7 by 3 sounding and a neighbourhood of 13 by 3 soundings respectively.

The first part of the procedure is the elimination of gross blunders. This is achieved by checking if each sounding's depth lies between a minimum and maximum depth and if its cross track distance is smaller than a maximum. If a sounding does not meet any one of these criteria it is tagged as an outlier. The minimum depth, maximum depth and maximum cross track distance are user defined.

The second part of the procedure is testing whether the depth of the evaluated sounding lies within an acceptable range. This range is determined by the weighted neighbourhood depth average. The depth of the evaluated sounding must lie within 10.0 m plus 1.5% of the weighted neighbourhood depth average. This means that if the weighted depth average is 100 m the evaluated sounding must lie between 88.5 m and 111.5 m. If a sounding does not meet with this criterion it is tagged as an outlier.

To ensure a meaningful weighted neighbourhood average the following steps are performed prior to the calculation of it.
In the first step each ping to be included in the neighbourhood is checked whether it is recorded within a 30 second time span of the current ping. This is to ensure that the pings are true neighbours and not separated by long distances.
The next step is to ensure that there is a minimum of 6 non-zero depths in the neighbourhood. If not then the evaluated sounding is tagged as an outlier.

In the third step weights are assigned to the neighbourhood soundings. The soundings along the current beam are given a weight of 2.0 and the soundings along the adjacent beams are given a weight of 1.0.
The fourth and final step is the rejection of neighbours suspected of being outliers. Since bad data would distort the weighted neighbourhood average and therefore could cause erroneous decisions based on this average they should be excluded from calculating the average. This is achieved by computing the average and standard deviation of all neighbours. Then each neighbour is compared to the average and if it deviates more than two times the standard deviation it is excluded from calculating the weighted neighbourhood average. By excluding a neighbour the average must be calculated again and the testing of the neighbours as a suspected outlier must be done over, this is thus an iterative step. After the exclusion of a neighbour the test of a minimum of 6 non-zero neighbours is repeated as well. Soundings excluded from calculating the weighted neighbourhood average are not tagged as an outlier.

4.3 Outlier detection using quantiles and 2D kriging

This process is developed by the French Naval Oceanographic and Hydrographic Service (SHOM) and consists of three algorithms and is performed on the raw data set, [Bisquay et al, 1998]. These three algorithms to detect outliers are based on the assumption that the sea floor bathymetry can be considered as continuous. The algorithms test the statistical consistency of a sounding with its neighbours. The tests are based on geostatistical techniques. However the decision to validate or invalidate a sounding remains by the operator and the detected outliers by the detection algorithms are used as a pre-selection of dubious soundings.

<u>The principles of the three algorithms:</u>

1. Testing the median of the depth increments between the evaluated sounding and its neighbours.
2. Defining an acceptance interval from the quantiles of the local distribution of the depth.
3. A cross validation is preformed in which the kriging error is compared with the kriging standard deviation.

<u>Process:</u>
The neighbourhood of the sounding to be evaluated consists of the soundings within a radius (user defined) of the evaluated sounding. The soundings belonging to the same ping as the sounding to be evaluated are excluded from the neighbourhood.

In the first algorithm the median of the absolute differences between the sounding to be tested and all its neighbours is computed. If this median is smaller then a detection threshold then the sounding is considered as valid.

In the second algorithm the local distribution of the depths in the neighbourhood of the sounding to be tested is computed. From this distribution the median and two quantiles are computed. If the sounding passes the following criterion it is accepted as valid.

$$Z_{0.5} + S_Q\left(Z_p - Z_{0.5}\right) \le z_{tested} \le Z_{0.5} + S_Q\left(Z_q - Z_{0.5}\right) \qquad \textbf{Equation 4.2}$$

in which:

$Z_{0.5}$ = the median

Z_p and Z_q = the quantiles of the order p and q for the local distribution

S_Q = the scale factor

The scale factor and the orders of the quantiles (p < 0.5 < q) are user defined.

In the third and last algorithm the kriging depth for every sounding is obtained from its neighbours. The kriging estimation also gives the estimate variance, which can be used to normalize the estimation error. The criterion to accept the sounding is then given by the next equation:

$$\left| z_{tested} - z_{tested}^{K} \right| < S_C \sigma_K \qquad \textbf{Equation 4.3}$$

in which:

z_{tested}^{K} = the kriging estimate of the depth

σ_K = the standard deviation of the estimation error and

S_C = a detection threshold

Thus if the absolute difference between the measured depth and the depth derived by kriging is smaller than the normalized estimation error (the criterion), it is considered as a valid sounding.

4.4 Outlier detection using quadratic surface fitting

This second procedure, developed by SHOM, tags a sounding as dubious and the decision to validate or invalidate is then done by an operator, [Debese et al, 1999]. This algorithm relies on local modelling of the seabed. It is assumed that at a given scale, the seabed topography can be modelled through a quadratic surface. Since this can not be achieved for the whole geographic area it is subdivided into sub-areas (i.e. into squared cells of the same size). For each sub-area a robust estimator is used to fit a quadratic model over the raw data. The difference between the measured depth and the depth estimated from the model is then used to test the consistency of each sounding with its neighbourhood.

The robust estimator used in this algorithm is the Tukey estimator. Unlike a least squares estimation it tries to accommodate the majority of data. The Tukey estimator is an iterative estimator whose rejection threshold of the soundings changes from one iteration to the next.
The algorithm requires two essential parameters, the inverse-sensitivity factor of the Tukey estimator (α) and the size of the cells (L). The inverse-sensitivity factor is used to compute the rejection threshold. The soundings with an absolute residual of α times greater than the median value of all the absolute residuals are seen as outliers. In the next iteration step they are also not taken into account. The size of the cells has to be chosen so that a quadratic surface can, with statistical verification, be fitted over the seabed of each cell.
The adaptive behaviour of the estimator has a drawback. Independently of the detection process it is necessary to introduce a global parameter defining the minimum magnitude of a valid sounding error. This parameter can be defined a priori from knowledge of the intrinsic depth uncertainty of the multibeam echo sounder.
The algorithm has two running modes. One in which each sounding is observed just one time, the fast running mode. The other mode is the cover mode where each sounding can be tested several times. In this mode each sounding is observed through several overlapping neighbourhoods.

4.5 Outlier detection using global and local variances

The algorithm that is introduced here is developed by the Center for Coastal and Ocean Mapping (CCOM) and the University of New Hampshire (UNH) together, [Tianhang Hou et al, 2001]. It is based on a buffer of 60 consecutive pings. This buffer is represented by a rectangular matrix (a_{ij}) that is populated by soundings that are indexed in the vertical by the ping number and indexed in the horizontal by the beam number. If one or more of the next three individual algorithms considers a point as an outlier, then it is tagged as one. All three the algorithms use the 60 ping buffer.

In the first algorithm an outlier is detected with the use of global and local variances. A sounding is detected as bad if its depth falls outside the range [mean depth - $s_1 \cdot \sigma^2_{test}$, mean depth + $s_2 \cdot \sigma^2_{test}$], where s_i is a factor in order to scale the threshold and σ^2_{test} is determined by the local and/or global variance according to the following logic:

$$\sigma^2_{test} = \begin{bmatrix} \sigma^2_{global} & if \quad \sigma^2_{global} \rangle \, \sigma^2_{local} \\ \\ \frac{1}{2}(\sigma^2_{global} + \sigma^2_{local}) & if \quad \sigma^2_{global} \langle \, \sigma^2_{local} \end{bmatrix}$$

The local variance is computed using the close neighbour soundings that are defined in a regular grid. In order to estimate the global variance, long spatial trends must be removed from the seabed representation. The de-trending is accomplished by using second order differentiation in both along track and across track directions.

$$b_{11} = a_{21} - \frac{a_{11} + a_{31}}{2} \text{ (along track)} \qquad \textbf{Equation 4.4}$$

$$c_{11} = a_{12} - \frac{a_{11} + a_{13}}{2} \text{ (cross track)} \qquad \textbf{Equation 4.5}$$

$$\sigma^2_{global} = \frac{0.5}{p-1}\sum_{k=1}^{p}(b_k - \bar{b})^2 + \frac{0.5}{p-1}\sum_{k=1}^{q}(c_k - \bar{c})^2 \qquad \textbf{Equation 4.6}$$

in which:

p = (ping number -2)·(beam number)
q = (beam number – 2)·(ping number)

For the de-trending Equations 4.4 and 4.5 are used. Then Equation 4.6 is used to compute the global variance.

In the second algorithm two sample variances are tested to detect outliers. The sounding to be tested lies at the centre of a fixed working window and the two samples H1 and H2 are defined from their close neighbour depth points as:

$$H1 = \begin{bmatrix} a_{i-1,j-1} & a_{i-1,j} & a_{i-1,j+1} \\ a_{i,j-1} & a_{i,j} & a_{i,j+1} \\ a_{i+1,j+1} & a_{i+1,j} & a_{i+1,j+1} \end{bmatrix}, \; H2 = \begin{bmatrix} a_{i-1,j-1} & a_{i-1,j} & a_{i-1,j+1} \\ a_{i,j-1} & & a_{i,j+1} \\ a_{i+1,j+1} & a_{i+1,j} & a_{i+1,j+1} \end{bmatrix}$$

Then a test variable G is constructed using:

$$G = \frac{\frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3}(a_{i,j} - \overline{a_{H1}})^2}{\frac{1}{8}\sum_{i=1}^{3}\sum_{j=1}^{3}(a_{i,j} - \overline{a_{H2}})^2} \qquad \textbf{Equation 4.7}$$

If this test variable is greater than 5.73 the tested sounding is considered an outlier.

The third algorithm is used to detect an erroneous ping. The first step is to re-arrange the close neighbour matrix LL into three matrixes L1, L2 and L3.

$$LL = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_{2_1} & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}; \; L1 = \begin{bmatrix} a_1 & a_2 & a_3 \\ c_1 & c_2 & c_3 \end{bmatrix}; \; L2 = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}; \; L3 = \begin{bmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

Then, the standard deviations of the subsets L1, L2 and L3 are computed based on the average depth of the subset L1. The standard deviations are computed by the succeeding:

$$S_i^2 = \frac{1}{5}\sum_{k=1}^{6}(Li_k - \overline{L1})^2 \quad \text{for} \quad i = 1 \ldots 3 \qquad \textbf{Equation 4.8}$$

If the standard deviations of $\frac{S_2^2}{S_1^2}$ and $\frac{S_3^2}{S_1^2}$ are both greater than a constant K a significant difference is detected between the middle row and the other two rows. This means that the sounding under evaluation b_{2_1} must be tested further. In the next step it is tested if the difference between the depth of sounding b_{2_1} and the average depth of subset L1 is greater than a criterion (Equation 4.9). If all three the conditions are satisfied, then the point under evaluation is detected as a bad ping and therefore is considered as an outlier.

$$(b_{2_1} - \overline{L1}) > S_{LL} \qquad \textbf{Equation 4.9}$$

4.6 Outlier detection using a Kalman filter process

This method is also developed by the CCOM and UNH together, [Calder et al, 2001]. This procedure is however based on a grid of estimation nodes. This grid is usually a regular grid but can be irregular. At each node the depth and its uncertainty is estimated. This allows for the possibility to combine data from multiple surveys, multiple instruments or multiple passes with the same instrument.

Sounding depths are augmented by an estimate of horizontal and vertical accuracy using a MBES (MultiBeam Echo Sounding) error model. They are then offered for incorporation into the grid nodes around the location of the sounding. Nodes close to the location of the sounding accepts the depth estimate at almost the computed accuracy. Those further away accept the depth estimate

but increases the uncertainty according to the distance. The operator gives the rate of dilution of the accuracy.
At a node the soundings are integrated to update the current depth and uncertainty estimates using an optimal Kalman filter. To ensure a decent robustness a re-ordering stage is included prior to the incorporation of the Kalman filter. This allows for a delay of those observations about which uncertainty is to great to use until information of less controversial data allows for a reliable process for the remaining outliers.

At each node an estimate of the depth and the variance associated with that estimate is maintained. As a new depth estimate arrives, the current estimate is used as a priori information, the current state of knowledge about the depth and the prediction uncertainty is updated with the new information. This update scheme is formalized in the Kalman filtering process.
The Kalman filter balances the error associated with new data and its current confidence in the estimate. Thus, if the input variance is of much lower variance than the estimate, then it is given a significant weight. Conversely if the input variance is of much lower value than the estimate it is mostly ignored. As more and more data is integrated the estimates of the depths becomes more and more accurate. This leads to an easy recognition of outliers since they are given low weighting as they are incorporated into the depth estimate. Errors or outliers in the early estimation sequence are still a concern. The base of the problem is one of initialisation. The typical initialisation value for the variance is 0.10^3 m ($\sigma_0 = 0.10^3 m$) resulting in an initially naïve filter concerning outliers. To protect the filter a median ordered pre-filter queue has been implemented, which effectively permutes the input sequence so that soundings, which are suspiciously high or low are delayed until later in the sequence at the node.

4.7 Discussion

One aspect of the problem leading to this master thesis is that the current outlier detection algorithm of Fugro Intersite B.V., which is cross validation applied to all soundings ping-wise, has in certain cases the tendency to ‘tag’ soundings that represent small objects, such as pipes, as outliers. This sometimes leads to removing a part or the whole pipe from the data set which is not the purpose since the soundings are not outliers.
As previously mentioned all the information about the above described outlier detection algorithms are obtained from articles, with one exception where the information was obtained from a master thesis. The algorithms can consequently only be judged based on the limited information that was available in the articles. There is nothing mentioned about the performance or lack of performance of the algorithms when a narrow pipe is present in the data set. The amount of information present is also nearly always insufficient to recreate the algorithm in order to test it using data containing a pipe. This process would also be too time consuming. It is thus not possible to judge whether the above outlier detection algorithms are adequate.

As a result a new algorithm is developed containing various features present in some of the described algorithms. These used features are:

- *Applying thresholding as a first step.*
  The usage of a thresholding prior to further testing of outliers is chosen because in almost all the algorithms a form of thresholding is done prior to further testing. It is furthermore done because if neglected then the gross blunders will influence the succeeding steps (the computation of the covariance function). It is also a simple and fast method of detecting gross blunders.

- *The usage of a number of neighbours*.
  The usage of a number of neighbours is another aspect that is used in all the above algorithms. Not all the algorithms define the number of neighbours explicitly but if defined the number of neighbours used vary per algorithm, from 4 neighbours to 39.
- *The usage of kriging in combination with cross validation*.
  All but the last procedure (Section 4.6) determine an outlier by comparing the measured depth by a computed depth and are thus applying cross validation. The manner in which the computed depth is calculated varies per procedure. Kriging is used in two procedures (Sections 4.1 and 4.3), but in different ways. In Section 4.3 it is done for an area determined by a radius (2D) and in Section 4.1 it is done for one ping only (1D). Since the method of Fugro Intersite B.V. is based on the procedure of Section 4.1 it is known to have drawbacks. Therefore kriging is done in two dimensions. Kriging is chosen because it is possible to determine how good the estimated (calculated) depth is which can then be taken into account when cross validating. Kriging also offers the possibility of quality control.

5 The 2D outlier detection algorithm

The algorithm is designed to process data that is acquired by a calibrated multibeam system and which are already corrected for errors that occur during the survey. The data is thus corrected for the attitude (roll, heave, pitch and heading (yaw)), the tide and other systematic errors. The algorithm is based on cross validation with the aid of kriging and a covariance function determined from the measured depths of the data. Cross validation, kriging and the covariance function have been discussed in Chapter 3.

The algorithm

The algorithm basically consists of six steps:

1 Thresholding
2 Reading the data into a buffer
3 Determining a covariance function
4 Determining the neighbours of a sounding
5 Calculating a predicted depth for a sounding
6 Testing the absolute difference between the measured depth and the predicted depth against a test criterion

Ad 1 Thresholding

This step is a preliminary step and done before running the actual 2D outlier detection algorithm. Multibeam data usually contains some soundings that where reflected by an air bubble, fish or suspended debris. These soundings do not represent the seabed. They are in fact much shallower than the seabed. It is also possible that a single sounding is much deeper than the seabed.
In this step these gross blunders are detected and removed from the data set through thresholding. In Figure 5.1 an illustration of thresholding is given. The removal of the gross blunders is necessary because they distort the calculation of a covariance function.
The threshold values are determined a priori by the user. If the minimum depth $Z_{\min}$ and maximum depth $Z_{\max}$ of the surveyed area is known, then all depths deeper than $Z_{\max}$ or shallower than $Z_{\min}$ are considered as gross blunders and removed from the data set. They are actually written to a separate file. A sounding is defined as a gross blunder if it is situated at more than two times the standard deviation of the depths referenced to the mean depth [Bottelier, 1998]. If the a priori threshold values are not known threshold values of the mean depth plus or minus two times the standard deviation of the depths is used.

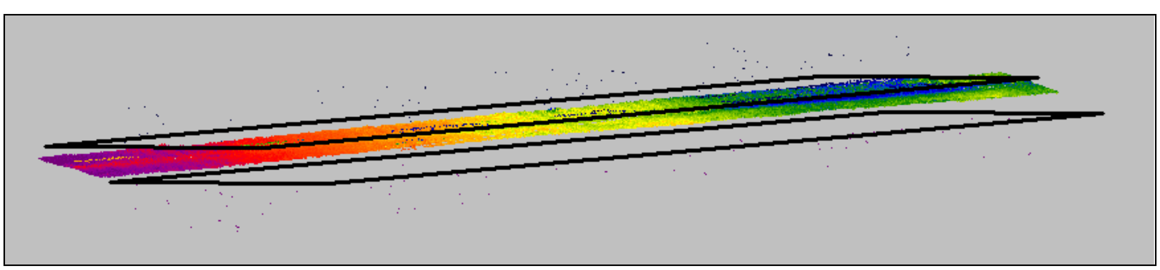

Figure 5.1 Thresholding as shown from a side perspective of a multibeam data set

Ad 2 Reading the data into a buffer

The second step is to read the data into a buffer so that only a small portion of the whole data set is processed at once. The size of the buffer is variable, since the number of pings that are read into the buffer is configurable. For example, the data can be processed per ping, per 10 pings or per 100 pings. It is in addition possible to vary the number of beams that is read into the buffer, thus for instance creating buffers of 10 pings by 10 beams or 50 pings by 30 beams.
The variability of the buffer size is also done in order to keep the process open for real-time application and to be able to compute a covariance function in a reasonable time.

Ad 3 Determining a covariance function

A covariance function is determined for all the soundings present in the buffer (after removal of the gross blunders). The covariance function is established as described in Chapter 4, thus firstly an empirical covariance function is determined which is smoothed with a moving average operator and from the resulting smoothed empirical covariance function an analytical function is established.
During the determination of the empirical covariance function a distance interval (*ds*) and discrete (s) values, at which covariance values must be calculated, are needed. The distance interval is usually the average separation distance of the soundings, thus the mean distance of all the distances *a* in Figure 5.2. In multibeam data the average separation distance of soundings in a ping and the average separation distance of soundings with the same beam number can differ significantly as can be seen in Figure 5.3. If the average separation distance of all the soundings is taken as the distance interval then the covariance function would have a jagged appearance. According to [Isaaks and Srivastava, 1989] if the maximum of the two (main) separation distances (the separation distances of soundings in a ping and soundings with the same beam number) is used then the covariance function would be smoother. The separation distance (*ds*) is thus the maximum of the average separation distance of either the pings or the beams. If the sampling pattern resembles the one shown in Figure 5.3, the separation distance of the soundings with the same beam number must be taken.
The discrete values (*s*) are multiples of the mean separation distance (*ds*), beginning with zero. The maximum value of *s* is about the maximum distance between all the soundings: if *s* exceeds the maximum distance there would not be any data points separated from each other by *s* and therefore no covariance value at distance *s*.

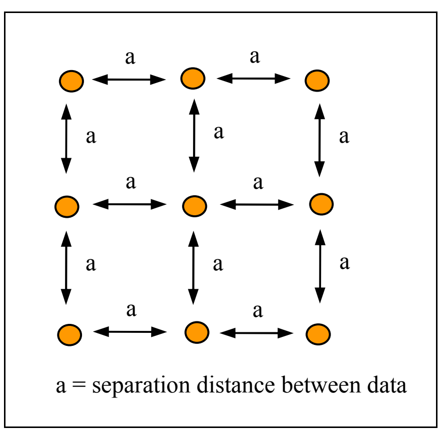

Figure 5.2 Illustration of separation distances

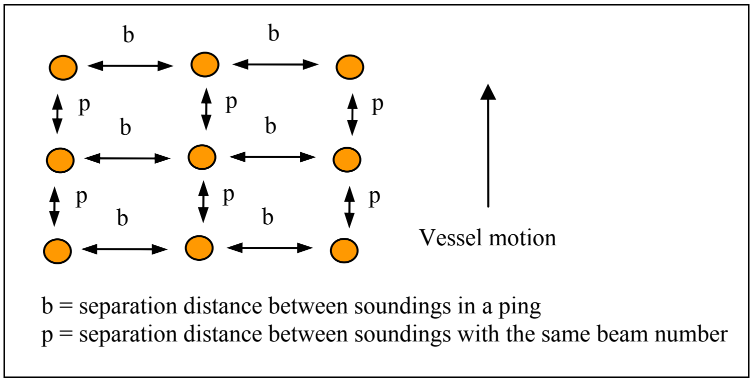

Figure 5.3 Illustration of separation distances for multibeam data

After the determination of the *ds* size and the maximum *s* value the empirical covariance function C_{n_s} is calculated using the succeeding equation.

$$C_{n_s}(s) = \frac{C_{1,n_s}(s)}{C_{1.n_s}(0)} C(0) \qquad \textbf{Equation 5.1}$$

in which:

$$C_{1,n_s}(0) = \frac{1}{2n_s} \sum_{i=1}^{n_s} (dZ_i^{\ 2} + dZ_{i'}^{\ 2})$$

and

$$C(0) = \frac{1}{N} \sum_{i=1}^{N} dZ_i^{\ 2} \approx \sigma^2$$

To decrease the irregular tendency, this empirical covariance function is smoothed with a moving average filter of five points:

$$\widetilde{C}_i = \frac{C_{i-2} + C_{i-1} + C_i + C_{i+1} + C_{i+2}}{5} \qquad \textbf{Equation 5.2}$$

Then in order to estimate an analytical covariance function as defined by Equation 5.3 [P.D. Bottelier, 1998] the zero crossing and correlation length of the empirical covariance function is determined. The zero crossing (*d*) is the distance at which the empirical covariance function crosses the covariance value zero for the first time, C(*d*) = 0. The correlation length (ξ) is the distance at which the value of the empirical covariance function is equal to half the value of the empirical covariance function at distance zero for the first time, C(ξ) = 0.5·C(0). In Figure 5.4 an example of the empirical covariance function together with the smoothed empirical and analytical covariance function is given.

$$C(s_{ij}) = C(0) \cdot (1 - f) \cdot e^{-f} \qquad \textbf{Equation 5.3}$$

in which:

$$f = \left(\frac{s_{ij}}{d} \right)^{\kappa}$$

and

$$\kappa = \frac{\log(0.3149)}{\log\left(\frac{\xi}{d}\right)}$$

in which:

s_{ij} = horizontal distance between two soundings
d = zero crossing of the covariance function
κ = curvature at C(0)
ξ = correlation length
C(0) = variance belonging to distance zero

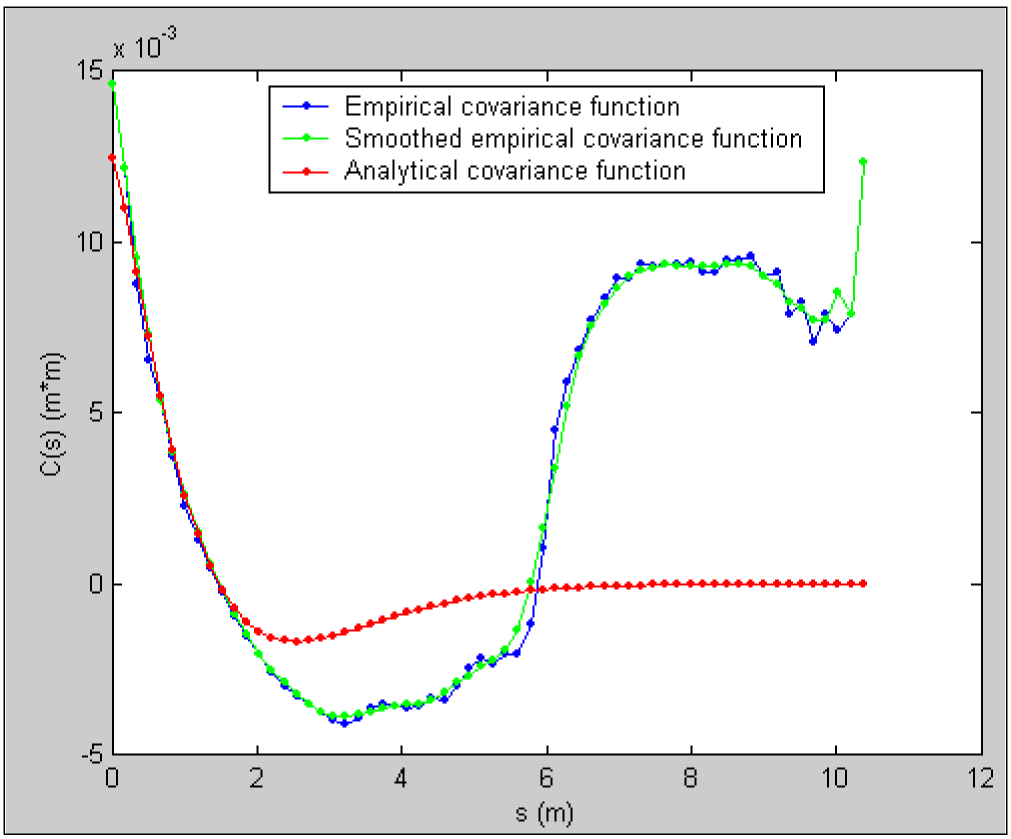

Figure 5.4 Example of an empirical, smoothed empirical and analytical covariance function

The covariance function can also be used to estimate the point noise (σ^2_{noise}) which is the random deviation (random error) present in each depth measurement. For multibeam data sets the maximum point noise is about 5 cm. A reliable estimate of this point noise is 90% of the difference between the C(0) and C(1·s) values, [Bottelier, 1998]:

$$\sigma_{noise} \cong \sqrt{0.9 \cdot (C(0) - C(1 \cdot s))} \qquad \textbf{Equation 5.4}$$

If a covariance function is calculated for a buffer with 20 pings by 60 beams it will differ from a covariance function calculated for a buffer of 20 pings by the first 30 beams of the ping. The covariance function for the other 30 beams by 20 pings will differ as well. The covariance function thus differs for each buffer and needs to be calculated for each buffer separately.

Ad 4 Determining the neighbours of a sounding

In this step the neighbouring soundings for each sounding is selected. In order to calculate a predicted depth it is necessary to define the neighbouring soundings that are to participate in the cross validation. It is possible to define the neighbours as, all the soundings that are within a radius *r* of the sounding to be cross validated, or just as the *m* closest neighbours. In this algorithm the last option is taken in combination with four mandatory neighbours.
Since the purpose of this algorithm is cross validation in two directions the mandatory neighbours are to ensure that the neighbours are not only located in one direction e.g. only in one ping or with sounding of only one beam number. The first two mandatory neighbours are the soundings which have the same ping number plus or minus one. The other two are the soundings with the same beam number plus or minus one. It is necessary to explicitly build this demand into the process because multibeam data can have a sampling structure in which the separation distance between soundings in a ping can be e.g. ten times greater then the separation distance between consecutive pings.

The closest neighbours have to fall within a radius (which is determined by the operator) of the cross validated sounding. In order to determine if a sounding falls within this radius not all the horizontal distances between the cross validated soundings and the other soundings in the buffer is calculated. First a bounding box is 'drawn' around the cross validated sounding. Of all the soundings that fall into the bounding box the distances to the cross validated sounding is calculated and determined if it is smaller than the radius. If the distance is smaller than the radius the sounding is a neighbouring sounding candidate.

In the process it is possible to define the number of neighbours manually, with a minimum of four neighbours (the mandatory neighbours). This means that if six neighbours are required the four mandatory neighbours are determined first and then the two closest (different) neighbours. An example is given in Figure 5.5. It is possible that one or more of the mandatory neighbours do not exist (e.g. at the edges of the buffer) or has been thrown away in a previous step. If that is the case then six neighbours are still used, only now e.g. three mandatory neighbours and three closest neighbours are used.

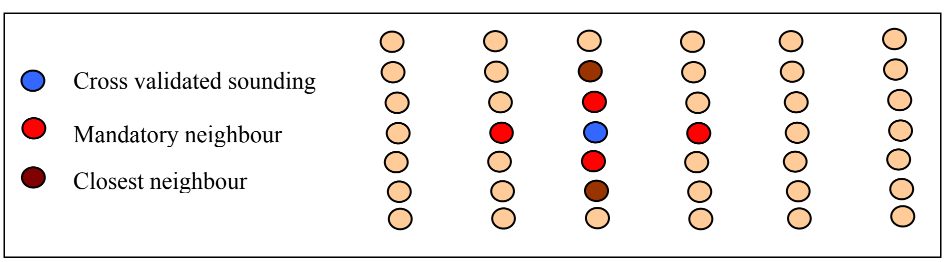

Figure 5.5 Six neighbours for cross validation (top view with 7 pings and 6 beams)

Ad 5 Calculating a predicted depth for a sounding

Kriging is used for the calculation of a predicted depth. This interpolation method is chosen because it is a best linear unbiased estimator. It is possible to calculate the prediction error (Equation 3.6) which can be used as a quality indicator for the estimation. Kriging uses a covariance function to determine the weights of the neighbours. This is another positive characteristic since in a covariance function the variation of the seabed is, up to a certain extent, taken into account.
After the neighbours have been determined in step four it is possible to calculate the weight of each neighbour with the aid of the following equation.

$$\begin{pmatrix} \omega_{P1} \\ \omega_{P2} \\ \vdots \\ \omega_{PQ} \\ \mu \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1Q} & 1 \\ C_{21} & C_{22} & \cdots & C_{2Q} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{Q1} & C_{Q2} & \cdots & C_{QQ} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} C_{P1} \\ C_{P2} \\ \vdots \\ C_{PQ} \\ 1 \end{pmatrix} \qquad \textbf{Equation 5.5}$$

$$\boldsymbol{\omega} = \mathbf{C}^{-1} \cdot \mathbf{D}$$

in which:

$C_{Pi} \equiv C(Pi)$

$C_{ij} \equiv C(s_{ij})$

μ = Lagrange multiplier

The values in the matrices C and D are calculated with the aid of the analytical covariance function determined in step three. With the weights known the predicted depth is calculated according to the successive equation.

$$d\hat{Z}_P = \sum_{i=1}^{B} \omega_{Pi} \cdot dZ_i \qquad \textbf{Equation 5.6}$$

in which:

$d\hat{Z}_P$ = predicted depth difference for sounding P

ω_{Pi} = weight of sounding *i*

dZ_i = depth difference of sounding *i*

B = number of neighbours

In this step the prediction error is calculated as well and is done according to the following equation.

$$\sigma^2{}_{prediction} = C(0) - D^T\omega = C(0) - D^T C^{-1} D \qquad \textbf{Equation 5.7}$$

Ad 6 Testing the difference between the measured depth and the predicted depth against a test criterion

The difference between the measured depth and the predicted depth is determined. Then in order to take the point noise and the prediction error into account when testing the absolute value it is dived by $\sqrt{\sigma^2_{noise} + \sigma^2_{prediction}}$. If this value exceeds the test criterion then the sounding is considered an outlier and removed from the data set. The actual test to determine if a sounding is an outlier is given in the following equation.

$$\frac{\left|Z_{measured,i} - Z_{predicted,i}\right|}{\sqrt{\sigma^2_{noise} + \sigma^2_{prediction}}} > \kappa_1$$ **Equation 5.8**

This test is actually equivalent to the *w*-test known in the mathematical geodesy [Teunissen, 2000, *Testing theory*].

$$w = \frac{v_k}{\sigma_{v_k}}$$ **Equation 5.8**

in which:

v_k = predicted residual

σ_{v_k} = variance of the predicted residual

The predicted residual is the same as the difference between the measured depth and the predicted depth. The variance of the predicted residual is the variance of the soundings (σ_{noise}) plus the variance of the prediction error ($\sigma_{prediction}$), [Teunissen, 2001, *Dynamic data processing*].

The test criterion is based on the assumption that the differences are normally distributed. A confidence level of 5% is used, which yields, using a Gaussian distribution function, a critical value of 1.96. This means that 5% of the data may be detected as outliers while they are not outliers.

The soundings are tested in a specific order. The first sounding tested is the sounding with beam number one of the first ping. Then the sounding with beam number two in that ping is tested, etc. until the last sounding in the ping is tested. Subsequently the soundings of the second ping are tested in the same order, followed by the third ping and so on. A sounding is removed from the data if it was considered an outlier during the cross validation process. The removed sounding does not participate in the testing of the following soundings, but it did participate in the testing of some soundings previous to its own testing.

6 Tests & Results

In order to evaluate the newly developed procedure, which will be called the 2D outlier detection algorithm, numerous tests have been executed using four different data sets. Usually a ground truth is used in order to evaluate the results of a procedure. But since there is no ground truth available for the data sets used in the tests the results have been inspected visually and compared to the results of a reference method, the 1D outlier detection algorithm.
The characteristics of the four data sets can be found in Section 6.1 while in Section 6.2 a review of the decisions made during the development of the algorithm will be given. A description of the major tests will be discussed in Section 6.3 and their results in Section 6.4.

6.1 Data sets

The data sets used to perform the tests differ from each other not only in the sea bathymetry but also in the multibeam systems used to acquire the data. In order to get an impression of the sea bathymetry the four data sets are visualized in Figures 6.1a to 6.1d using a 3 dimensional visualizer (Terramodel 9.6) that creates Digital Terrain Models (DTM's). The data used to generate the DTM's are corrected for the attitude, tide and other systematic errors.

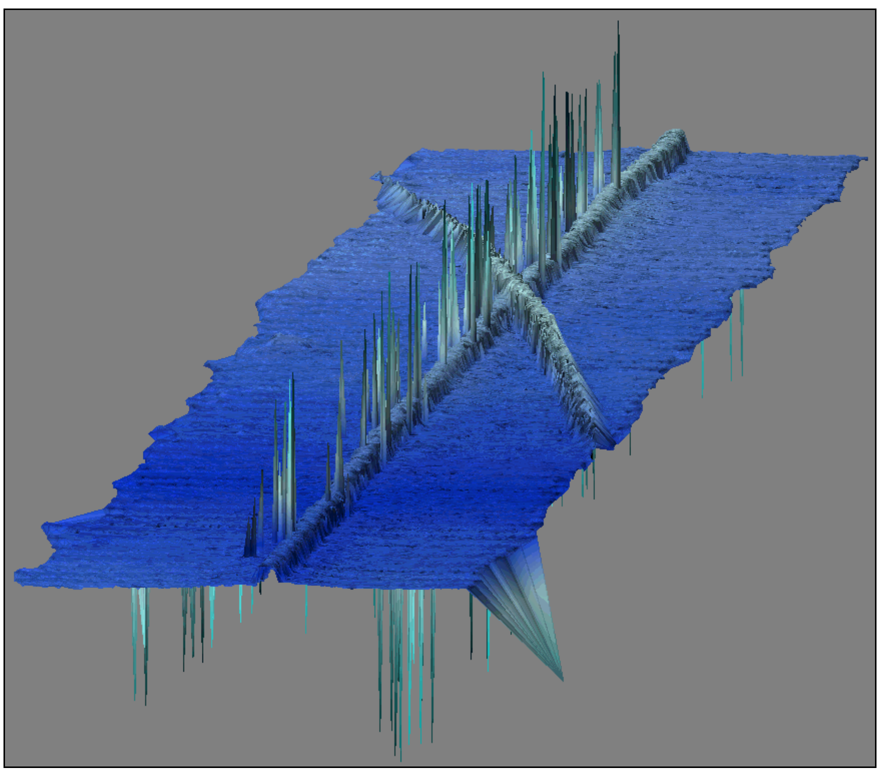

Figure 6.1a Overview data set 1 before processing

The first data set has been acquired using a Reson Seabat 9001 multibeam system. This system has 60 beams and a ping rate of 7 pings a second. The average depth is around 145 meters and the distance between consecutive pings is approximately 0.03 cm and the approximate distance between two adjoining beams is 0.1 cm.

As can be seen in Figure 6.1a the first data set is a recording of two pipes crossing each other on a relatively flat bottom. Apart from the two distinctive pipes a lot of huge spikes are visible as well. These spikes are most probably gross blunders which should be removed by the thresholding.

The second data set has been acquired with the Reson Seabat 8101 multibeam system, which has 101 beams and a ping rate of 9 pings per second. Here the average depth is approximately 13 metres in the first half of the data set and then 23 metres in the second half. The approximate distance between consecutive pings is 0.13 cm and between two adjoining beams 0.45 cm.
The reason for this data set having a jump in the depth average is because there is a short steep slope in the middle of the data set, which can be seen clearly in Figure 6.2. In this data set there are no pipes present but it has been recorded with probably a defect beam. In the middle a belt of spikes is visible. This indicates a bad beam since it is only one sounding thick.

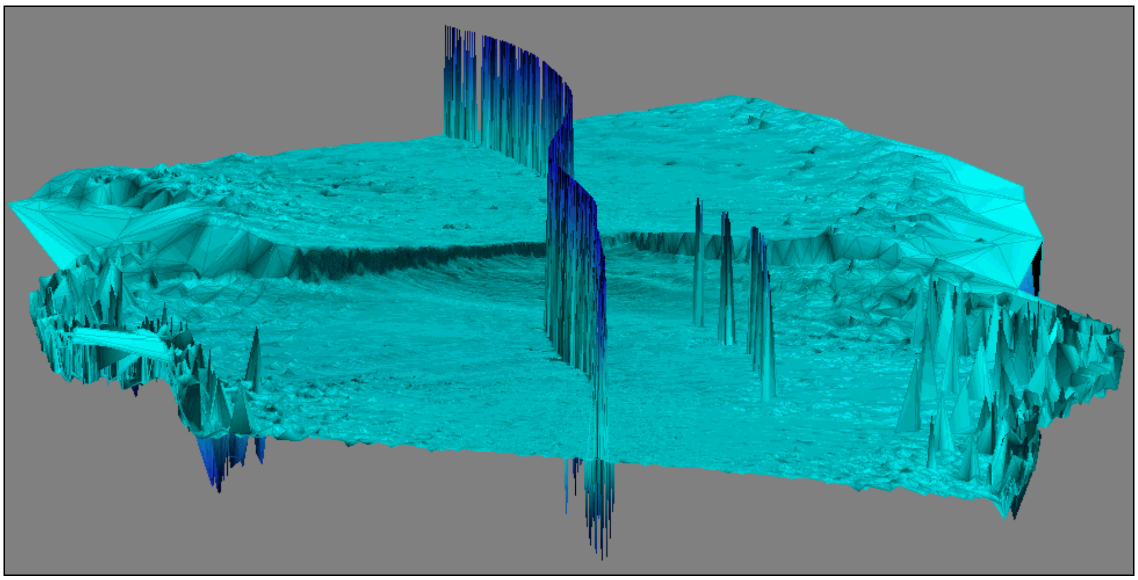

Figure 6.1b Overview data set 2 before processing

The third and fourth data sets have been acquired with the Reson Seabat 8125 multibeam system, which has 240 beams and a ping rate of 2 pings per second. The third data set has an average depth of 304 metres and an approximate distance of 0.14 cm between consecutive pings and 0.19 cm between two adjoining beams. As can be seen in Figure 6.1c there are yet again spikes present and one pipe in the middle. There is also a kind of plateau, called a template, in the middle over which the pipe runs. A template is a construction at the sea bottom connecting an infrastructure of pipes to an oil or gas field which is located in the earth.

The average depth of the fourth data set is 303 metres and the distance between two consecutive pings is approximately 0.12 cm and that of two adjoining beams 0.24 cm. In Figure 6.1d it can be seen that in this fourth data set there is also a pipe which runs over a template similar to the one in the third data set. The difference is that in this data set the pipe lies more to the left edge of the data set and that there is another bigger and more irregular kind of plateau present at the right side of the template.

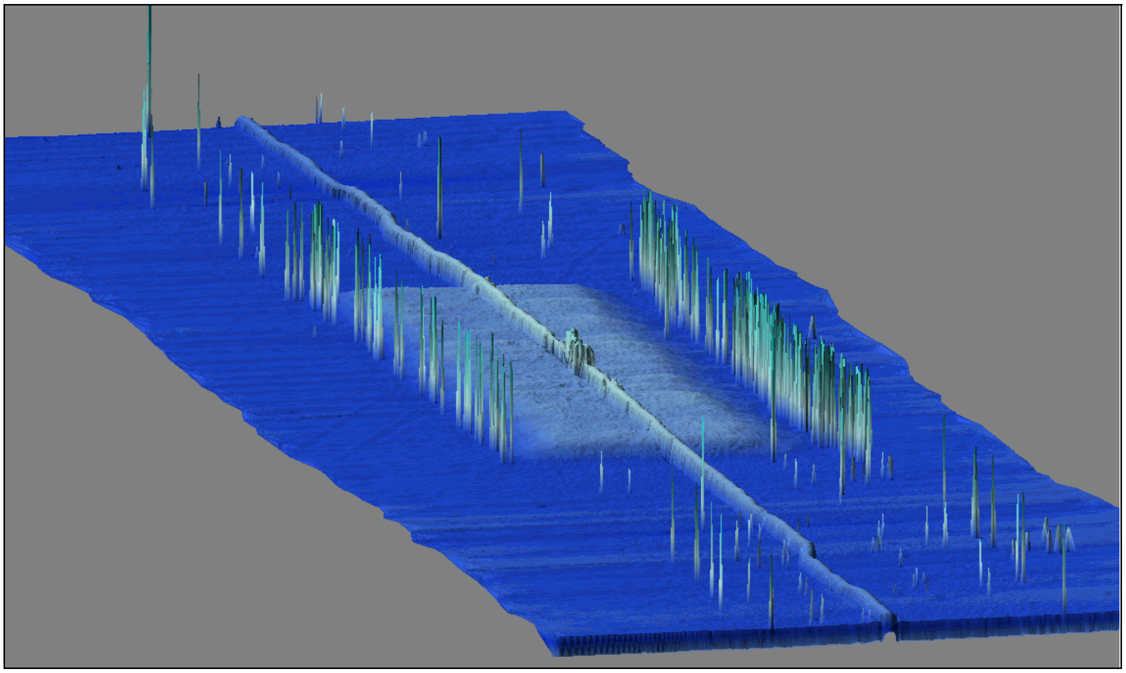

Figure 6.1c Overview data set 3 before processing

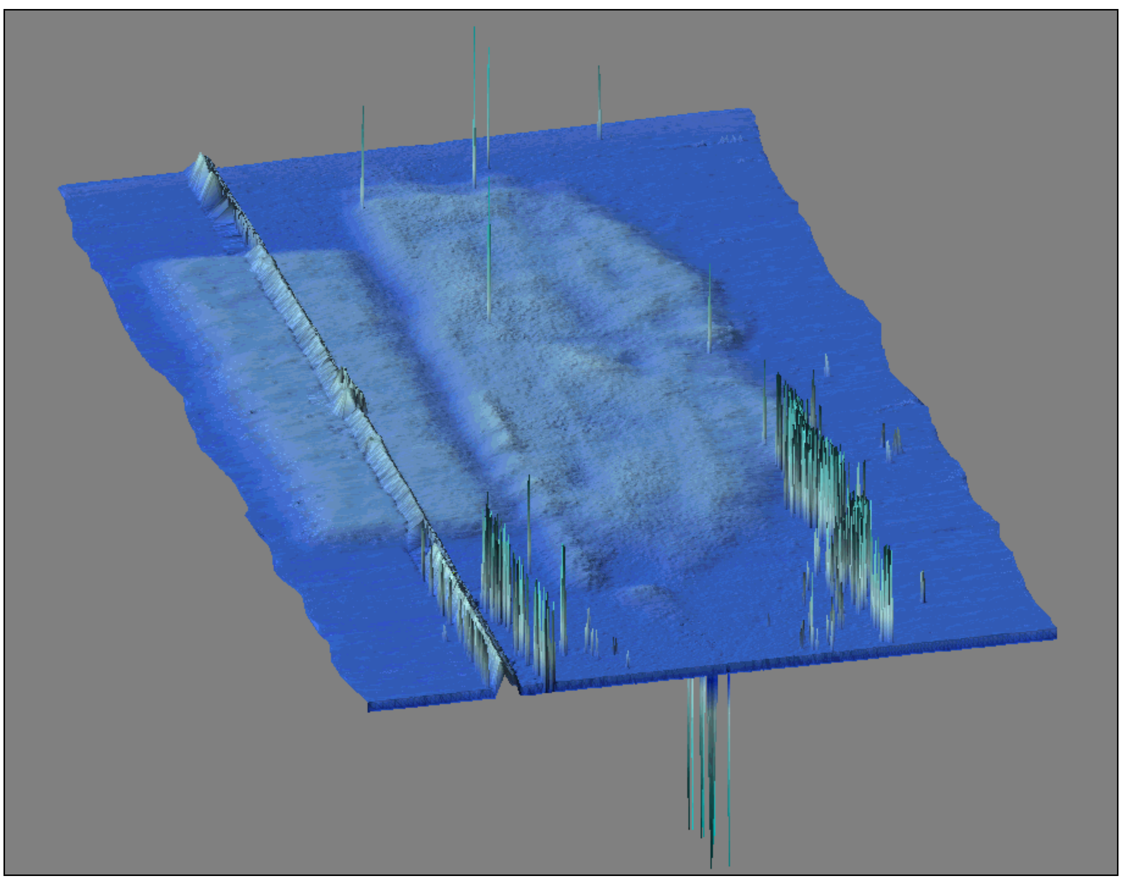

Figure 6.1d Overview data set 4 before processing

6.2 Development tests and results

During the development of the 2D outlier detection algorithm the problem arose on how to define the neighbours. This was an issue since if one were only to take the six closest neighbours it was possible that neighbours would only lie in one direction, as can be seen in Figure 6.2a. This problem occurred when the distance between two adjacent beams was much greater then the distance between two pings. This will likely happen as well if the distance between pings is much greater than the distance between two adjacent beams. Since the purpose is 2D outlier detection, the neighbours used must not lie in only one direction but should be forced to be located in two directions. This problem is eliminated by using at least the four mandatory neighbours as specified in Chapter 5. Figures 6.2b shows the neighbours used if four mandatory neighbours are used and two additional closest neighbours.

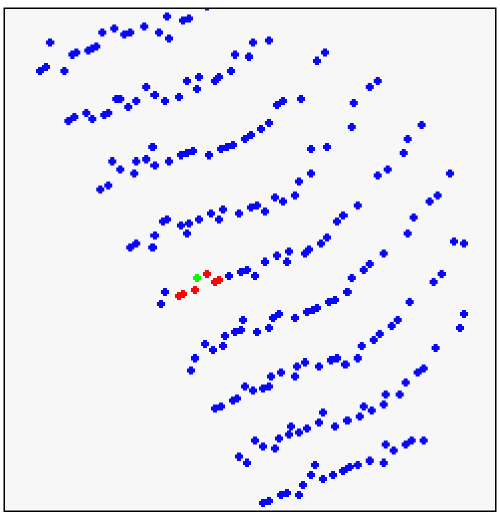

Figure 6.2a Six closest neighbours

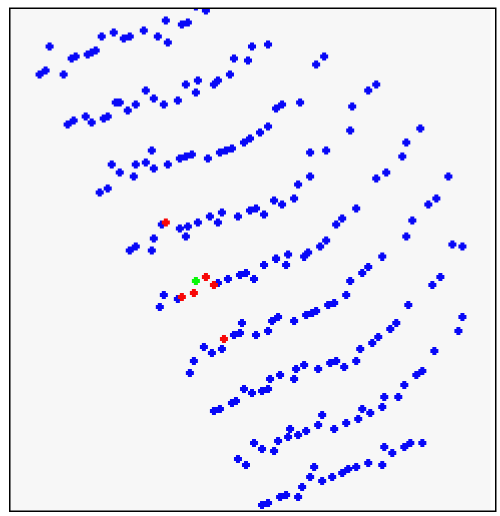

Figure 6.2b Four mandatory neighbours and two additional closest neighbours

Also during the development of the 2D outlier detection algorithm a query concerning the covariance function arose. While contemplating a method in which to reduce computing time the idea occurred that if the covariance function does not differ a lot for each buffer, it might be possible to calculate a covariance function for one buffer and use that covariance function for say the following five buffers. In order to determine if two covariance functions, in two consecutive buffers, are similar enough to each other the two parameters of the analytical covariance function, the zero passage (d) and the correlation length (ξ), have been compared to each other (see for a description of the parameters Chapter 3).

In Table 6.1 the values for the two parameters are given for three consecutive buffers of data set 1. These parameters do in fact lie close to each other but the problem is that it is not easy to predict what the impact of small changes of the two parameters is with respect to cross validation. Another aspect is that the point noise is different for each buffer and it influences the analytical covariance function as well. This implied that in order to determine if it is possible to use one covariance function for multiple buffers numerous tests would be needed and unfortunately there was not enough time to look into this properly.

Table 6.1 Values for the two parameters of the analytical covariance functions of three buffers with a buffer size of 50 pings (data set 1)

Zero passage (*d*) [m]	**Correlation length (ξ) [m]**
0.4870	1.6052
0.4803	1.5527
0.4891	1.4646

Another aspect that was tested during the development concerned the feasibility to split the number of beams read into a buffer. Thus instead of reading all e.g. 60 beams of 30 pings into one buffer, read beams 1 to 30 in one buffer and the beams 31 to 60 in another. In Figure 6.3 an example of buffers of 15 pings by 15 beams is given. A problem that then arises is that a pipe can now lie at the utmost edge of a buffer or even be split in half. It is also possible that only in a small corner of the buffer a piece of pipe will be present. These circumstances are not desirable since it probably increases the chance that the soundings representing the pipe will be tagged as outliers. This is because the covariance function has little to no knowledge that a pipe is present, the fact that there are soundings close to each other but with a greater difference in height is not accounted for. Therefore it was decided to deactivate the possibility to vary the number of beams read into a buffer. Instead the total number of beams per ping is used. Obviously this differs for different multibeam data acquisition systems.

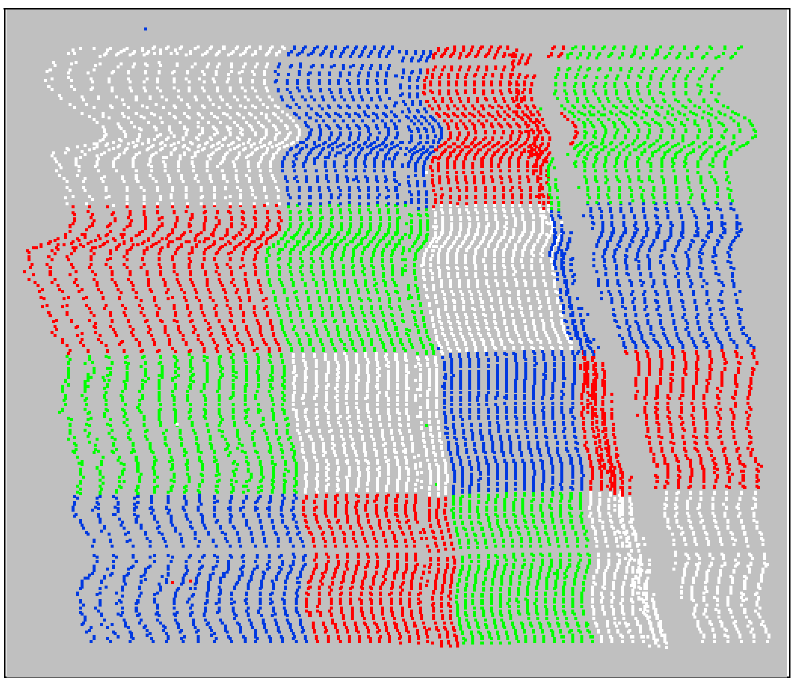

Figure 6.3 Example of buffers of 15 pings by 15 beams (data set 1)

6.3 Description of the major tests

Five tests per data set were performed using a 1D outlier algorithm. The used 1D outlier detection algorithm is the procedure developed by P.D. Bottelier whose concept is currently used at Fugro Intersite B.V. For testing purposes during this thesis this algorithm has been used without testing the first two and the last two beams in a ping. In the first five tests the only parameter which is varied is the test criterion. The number of neighbours is set fixed to four and is not adjustable by the user. The values for the test criterion that were used in the tests are 0.52, 0.84, 1.28, 1.96 and 2.57. These values are based on confidence levels and are in respective order 60%, 40%, 20%, 5% and 1%. A confidence level of 5% implies that 5% of the data may be detected as outliers when in fact they are not outliers.

Using the 2D algorithm, 17 tests have been performed on each data set. These tests were done in order to determine if this algorithm is better than the 1D outlier algorithm and to determine with which values for the parameters the procedure functions best. The parameters that were changed are: the number of neighbours used, the number of pings read into the buffer and the value for the test criterion.

The influence of the number of neighbours is checked by successively using 4, 6, 8, and 10 neighbours. This series of tests was conducted with a buffer size of five pings and again with a buffer size of ten pings. The test criterion was kept at 1.96.
The number of pings read into the buffer was evaluated by constantly using a test criterion of 1.96 and 6 neighbours. For each data set six tests were performed using in turn 5, 10, 15, 20, 30 and 50 pings per buffer.
In the last series of tests the test criterion was altered using the same values as for the 1D outlier detection algorithm, thus the values 0.52, 0.84, 1.28, 1.96 and 2.57. The number of neighbours was set to 6 and the number of pings to 50. These tests were only executed using the first three data sets.

The 1D and the 2D algorithm both use thresholding to detect gross blunders. It is for both algorithms a preliminary step and in both algorithms the threshold values can be given by the operator. In order to eliminate the influence of thresholding the same threshold values were used in both algorithms. In Figures 6.4a to 6.4d DTM's of the four data sets are given after thresholding. If compared to respectively Figures 6.1a to 6.1d it can be seen that the major spikes and thus gross blunders have been removed.

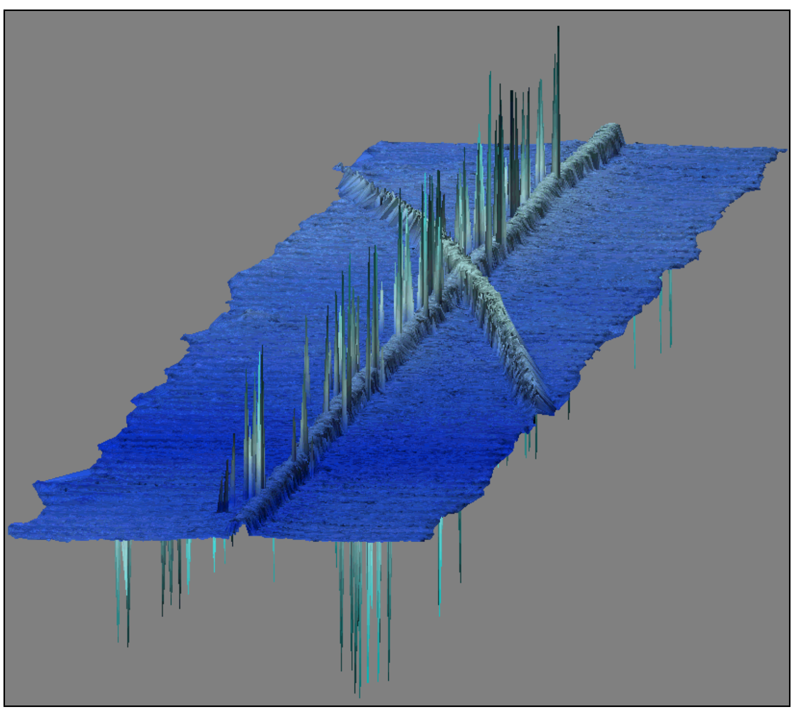

Figure 6.4a Data set 1 after thresholding

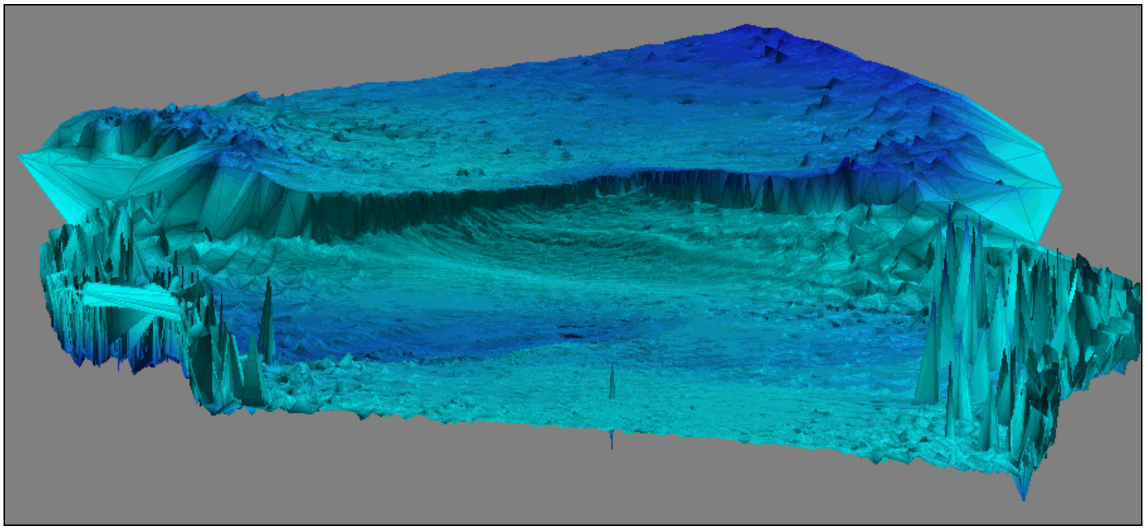

Figure 6.4b Data set 2 after thresholding

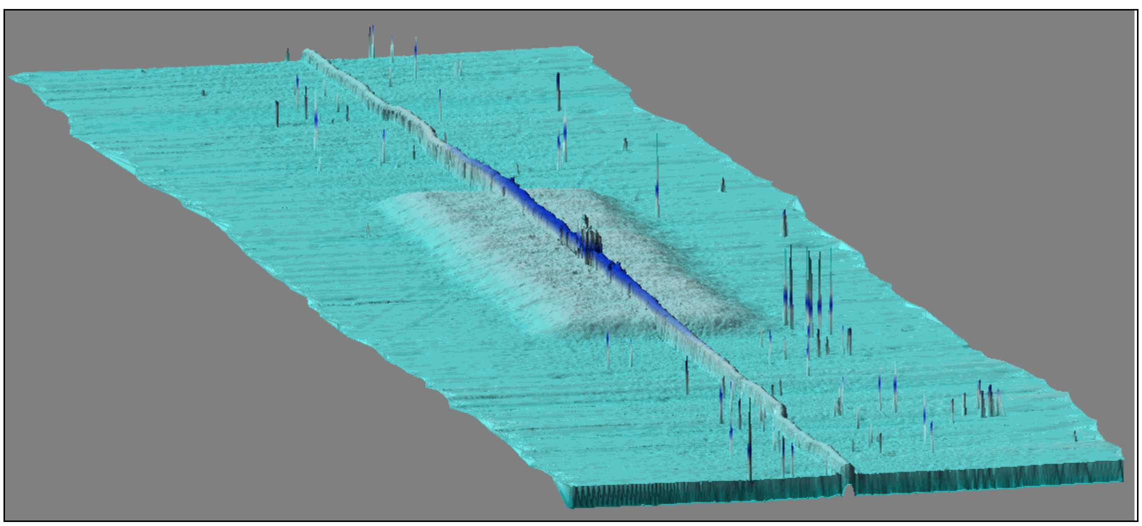

Figure 6.4c Data set 3 after thresholding

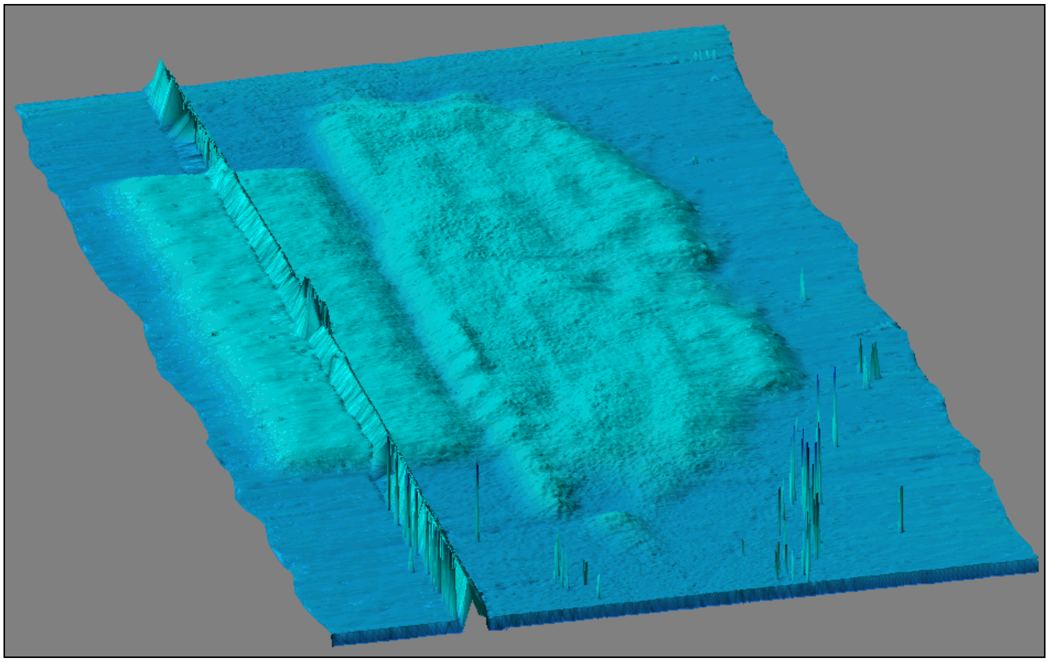

Figure 6.4d Data set 4 after thresholding

6.4 Results of the major tests

The purpose of the 2D outlier detection algorithm is to detect outliers. After visual inspection by human graphical interpretation of the first results it could be seen that the 2D algorithm detected some false outliers. This was also the case with the 1D algorithm. In order to determine if the amount of false outliers has decreased significantly, the number of outliers detected by the 2D algorithm is compared to that of the 1D algorithm. The results of the tests are furthermore inspected visually in order to establish if soundings that should have been detected as outliers have indeed been detected as outliers and to check if the detected outliers were located in approximately the same regions as the outliers that are detected by the 1D algorithm. As reference results the results from the 1D outlier detection algorithm, with a test criterion of 1.96 (confidence level of 5%) and four neighbours is used. The confidence level is set at 5% since it is (in the geodetic world) standard practise and the number of neighbours is set to four because this is the value used at Fugro Intersite B.V.

First the results of the reference 1D outlier detection algorithm test will be given then the results of the tests done with the 2D outlier detection algorithm will be given per varied parameter. The tests where the criterion values are changed for the 1D and 2D algorithms will be presented together at the end of this section.

The reference results

In Table 6.2 the reference results are presented numerically. The number of outliers and the number of accepted soundings are given in absolute numbers and in percentages of the number of soundings tested. The 1D algorithm used does not test the first two and the last two soundings in a ping thus the total number of outliers tested per data set will be less than with the 2D algorithm. In all the data sets except maybe the second data set, the first and last two soundings of a ping are generally not outliers.

Table 6.2 Reference results using the 1D algorithm

Data set	# soundings per ping	# outliers	% outliers	# soundings
1	60	3149	3.03	104040
2	101	3853	3.08	125245
3	240	4364	1.46	297958
4	240	1955	0.91	215557

Data set 1

From the 104,040 soundings tested 3,149 soundings were tagged as outliers. Inspecting the results graphically it can be seen that greater portion of the 'outliers' are situated at the edges of the pipes. In Figure 6.5 a top view of the outliers and accepted soundings of data set 1 using the 1D algorithm is given. These outliers are actually not outliers at all if one inspects them closer but they represent the pipes or the sea bottom adjacent to it. After a visual inspection of the results there are 21 soundings not detected as outliers while they are in fact outliers. This can clearly be seen when looking at the DTM of the soundings that were accepted.

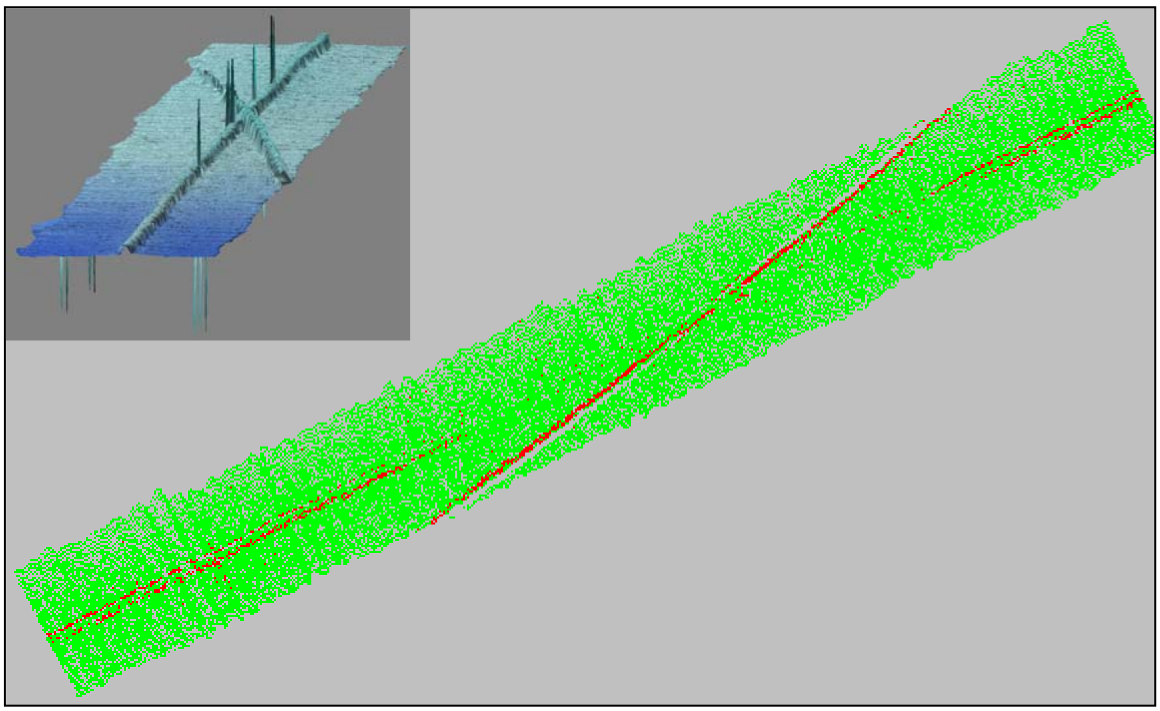

Figure 6.5 Top view of the accepted (green) and rejected (red) soundings inlaid with a DTM of the accepted soundings of data set 1 using the 1D algorithm

Data set 2

From the 12,5245 soundings 3,853 soundings where detected as outliers. As can be seen in Figure 6.6 most of the outliers are situated at the edges of the data set which is expected when one takes another look at Figure 6.4b (data set 2 after thresholding). For this data set it is difficult to see if the soundings that are detected as outliers are indeed outliers. It is also difficult to determine if all the outliers are indeed detected as outliers, but looking at the inlaid DTM in Figure 6.6 it is obvious that not al the outliers has been detected since there are still some spikes present.

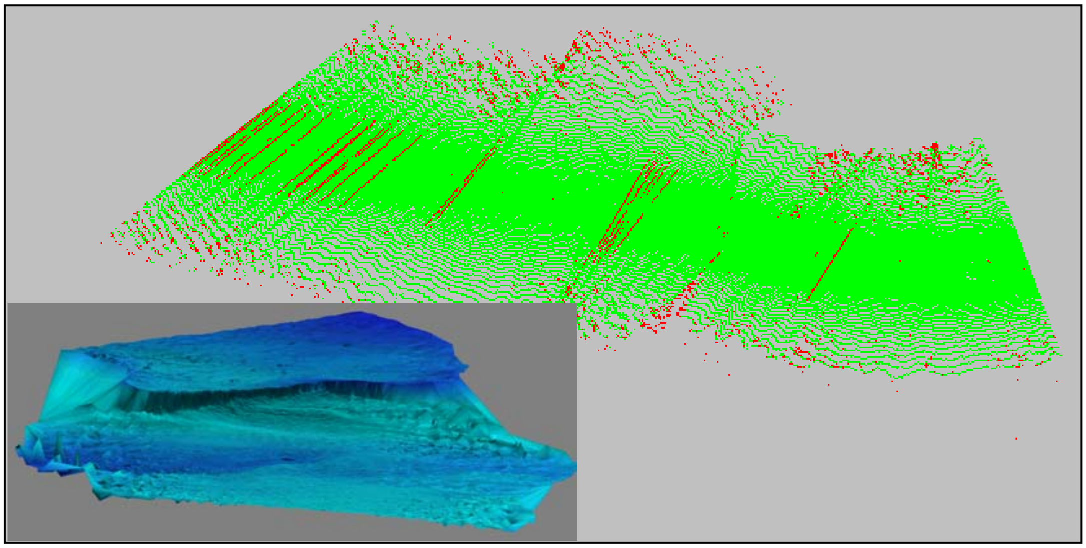

Figure 6.6. Top view of the accepted (green) and rejected (red) soundings inlaid with a DTM of the accepted soundings of data set 2 using the 1D algorithm

Data set 3

From the 297,958 soundings 4,364 soundings were detected as outliers. Again the majority of the detected outliers lie at the edges of the pipe as can be seen in Figure 6.7. After a closer examination of the results it could be seen that not all the outliers were detected. The still present outliers are represented by the spikes the inlaid DTM of Figure 6.7.

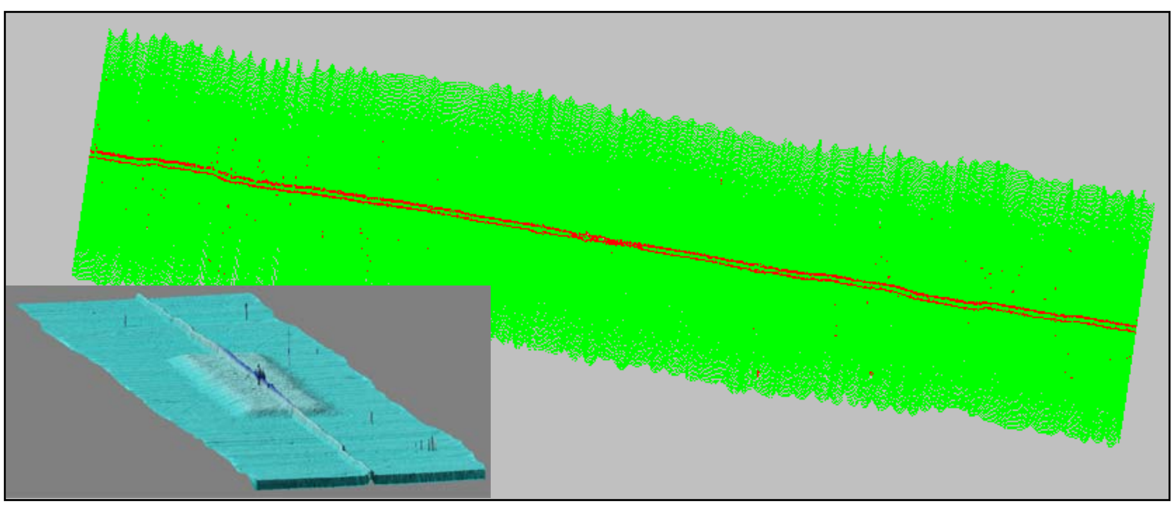

Figure 6.7 Top view of the accepted (green) and rejected (red) soundings inlaid with a DTM of the accepted soundings of data set 3 using the 1D algorithm

Data set 4
From the 215,557 soundings 1,955 soundings were detected as outliers. In Figure 6.8 it can be seen that once more the majority of the outliers were situated at the edge of the pipe. Consistent with the first three data sets there are yet again undetected outliers (see the inlaid DTM).

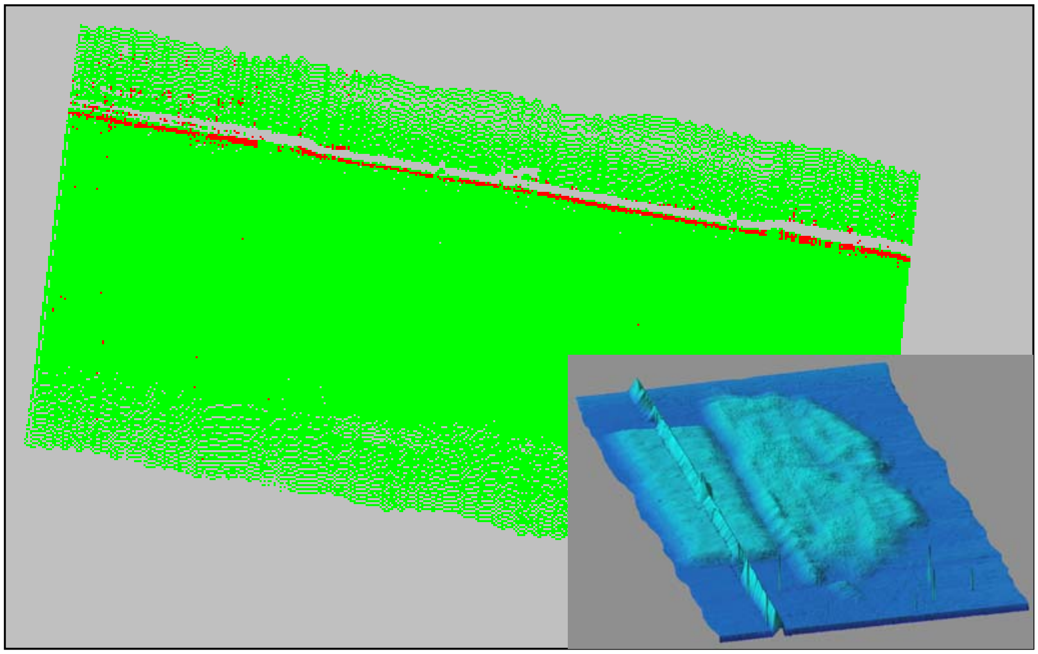

Figure 6.8 Top view of the accepted (green) and rejected (red) soundings inlaid with a DTM of the accepted soundings of data set 4 using the 1D algorithm

Number of neighbours

The number of neighbours used in the 2D outlier detection algorithm influences the number of outliers detected. But it is not possible to state 'that the more neighbours used the less outliers detected'. Looking at Table 6.3 four out of eight times the usage of four neighbours produces the least outliers but the other four times the most. It is thus not even possible to say that a number of neighbours gives the best results.

A possible explanation might lie in the manner in which the neighbours are defined. The mandatory neighbours are not necessarily the closest neighbours, see Figure 6.2b. The influence of the neighbours that are closer to the cross validated sounding on the predicted depth is thus not taken into account. Neglecting the closer neighbours may thus not be tolerable.

Table 6.3 Results of the tests concerning the number of neighbours used

Data set	# pings	# neighbours	# outliers	% outliers	# soundings
1	5	4	1346	1.21	111476
1	5	6	1525	1.37	111476
1	5	8	1514	1.36	111476
1	5	10	1538	1.38	111476
1	10	4	1167	1.05	111476
1	10	6	1403	1.26	111476
1	10	8	1394	1.25	111476
1	10	10	1415	1.27	111476
2	5	4	2312	1.77	130641
2	5	6	2114	1.62	130641
2	5	8	2097	1.61	130641
2	5	10	2059	1.58	130641
2	10	4	1956	1.50	130641
2	10	6	1793	1.37	130641
2	10	8	1746	1.34	130641
2	10	10	1745	1.34	130641
3	5	4	3769	1.24	303014
3	5	6	3432	1.13	303014
3	5	8	3293	1.09	303014
3	5	10	3466	1.14	303014
3	10	4	3599	1.19	303014
3	10	6	3238	1.07	303014
3	10	8	3084	1.02	303014
3	10	10	3296	1.09	303014
4	5	4	3053	1.39	219217
4	5	6	3486	1.59	219217
4	5	8	3455	1.58	219217
4	5	10	3418	1.56	219217
4	10	4	2869	1.31	219217
4	10	6	3492	1.59	219217
4	10	8	3479	1.59	219217
4	10	10	3403	1.55	219217

The number of pings read into the buffer

After looking at the results of the tests concerning the number of neighbours, it could be seen that the number of pings read into the buffer has a certain influence on the number of outliers detected. Table 6.4 gives the results of the tests where the number of pings read into the buffer is altered. The first three data sets clearly indicate that the more pings read into the buffer the less outliers are detected. The number of outliers detected (for the first three dataset) if a buffer size of 50 pings is used, is about one third of the number of outliers detected by the 1D algorithm. After inspecting the results visually it is clear that all the real outliers in data set 1 have been detected (see Figure 6.9) and all outliers except one or two small ones in data set 3. See Appendix A for DTM's of all the accepted soundings of all four data sets using the 2D outlier algorithm with 50 pings. In Appendix B the corresponding top views of the accepted and rejected soundings can be found.

Table 6.4 Results of the tests where the number of pings read into the buffer is altered

Data set	# pings in the buffer	# outliers	% outliers	# soundings
1	5	1525	1.37	111476
1	10	1403	1.26	111476
1	15	1398	1.25	111476
1	20	1231	1.10	111476
1	30	1183	1.06	111476
1	50	1064	0.95	111476
2	5	2114	1.62	130641
2	10	1793	1.37	130641
2	15	1739	1.33	130641
2	20	1645	1.26	130641
2	30	1513	1.16	130641
2	50	1478	1.13	130641
3	5	3432	1.13	303014
3	10	3238	1.07	303014
3	15	3124	1.03	303014
3	20	2967	0.98	303014
3	30	2808	0.93	303014
3	50	2538	0.84	303014
4	5	3486	1.59	219217
4	10	3492	1.59	219217
4	15	3585	1.64	219217
4	20	3567	1.63	219217
4	30	3510	1.60	219217
4	50	3554	1.62	219217

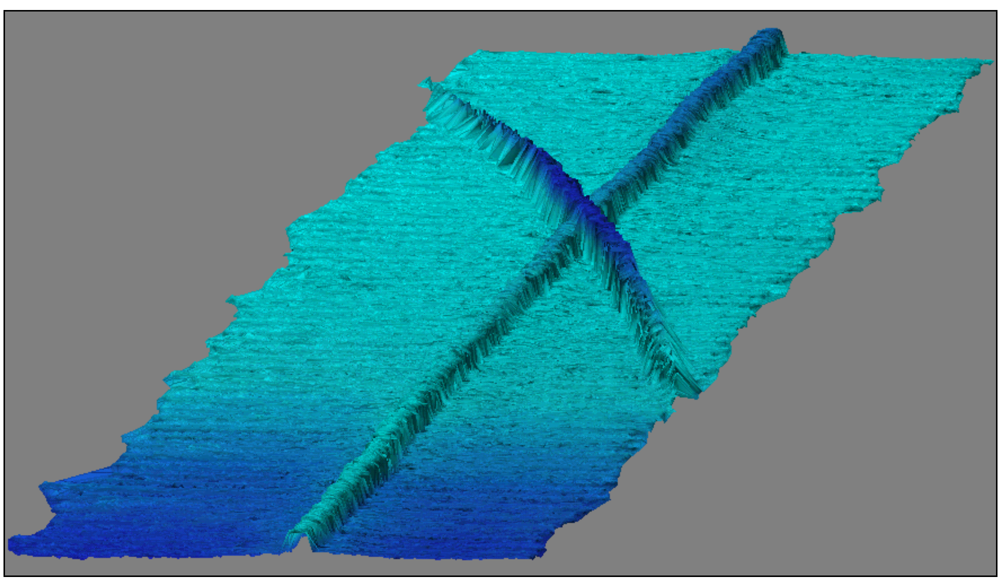

Figure 6.9 DTM of the accepted soundings of data set 1 using the 2D algorithm

The test criterion

Table 6.5 contains the results of the tests concerning the test criterion when using the 2D outlier detection algorithm. The chosen test criterion has a huge influence on the number of outliers detected. The greater the value for the test criterion the smaller the number of outliers detected. This is consistent with the results when varying the test criterion for the 1D outlier detection algorithm, see Table 6.6. It is also consistent with the theory which states that the smaller the confidence level, the smaller the number of falsely detected outliers. The down side of decreasing the confidence level is that the chance that true outliers are not detected increases. But for the 2D algorithm the number of undetected outliers is, if present, in the order of five when using a confidence level of 1%.

Table 6.5 Test results concerning the test criterion using the 2D outlier detection algorithm

Data set	test criterion	# outliers	% outliers	# soundings
1	0.52 (60%)	5141	4.61	111476
1	0.84 (40%)	2613	2.34	111476
1	1.28 (20%)	1800	1.61	111476
1	1.96 (5%)	1064	0.95	111476
1	2.57 (1%)	662	0.59	111476
2	0.52 (60%)	5445	4.17	130641
2	0.84 (40%)	3328	2.55	130641
2	1.28 (20%)	2184	1.67	130641
2	1.96 (5%)	1478	1.13	130641
2	2.57 (1%)	1150	0.88	130641
3	0.52 (60%)	10158	3.35	303014
3	0.84 (40%)	5335	1.76	303014
3	1.28 (20%)	3840	1.27	303014
3	1.96 (5%)	2538	0.84	303014
3	2.57 (1%)	1837	0.61	303014

Table 6.6 Test results concerning the test criterion using the 1D outlier detection algorithm

Data set	Test criterion	# outliers	% outliers	# soundings
1	0.52 (60%)	13758	13.22	104040
1	0.84 (40%)	8936	8.59	104040
1	1.28 (20%)	5955	5.72	104040
1	1.96 (5%)	3149	3.03	104040
1	2.57 (1%)	1724	1.66	104040
2	0.52 (60%)	16265	12.99	125245
2	0.84 (40%)	10051	8.03	125245
2	1.28 (20%)	6287	5.02	125245
2	1.96 (5%)	3853	3.08	125245
2	2.57 (1%)	2930	2.34	125245
3	0.52 (60%)	16956	5.69	297958
3	0.84 (40%)	9044	3.04	297958
3	1.28 (20%)	6386	2.14	297958
3	1.96 (5%)	4364	1.46	297958
3	2.57 (1%)	3187	1.07	297958

7 Conclusions & Recommendations

The developed 2D algorithm to detect outliers in multibeam data has been explained in Chapter 5. Chapter 6 showed the tests performed using this algorithm and the test results. This Chapter deals with the conclusions drawn from the test results in section 7.1 and the recommendations for further research in section 7.2.

7.1 Conclusions

The tests executed using the 2D algorithm are split up according to the three parameters tested: the number of neighbours used, the number of pings read into the buffer and the value of the test criterion. The conclusions concerning the influence of these three parameters on the performance of the algorithm will be given first. The overall conclusions on the performance of the algorithm will follow.

- *The influence of the number of neighbours used.*
  The number of neighbours used in the 2D algorithm has an influence on the results of the algorithm but has no apparent trend. The influence of the choice for the number of neighbours used is thus not clear.
- *The influence of the number of pings read into the buffer.*
  The number of pings read into the buffer has a significant influence on the performance of the algorithm. The more pings that are read into the algorithm the less false outliers are detected thus the better the results. If instead of five pings fifty pings are read into the buffer 30% less false outliers are detected. The number of true outliers not detected decreases as well.
- *The influence of the value of the test criterion.*
  The value of the test criterion influences the number of falsely detected outliers and thus the performance of the algorithm. The bigger the test criterion (the smaller the confidence level) the less false outliers are detected. This is according to the theory which says that the smaller the confidence level, the smaller the number of falsely detected outliers. In spite of what is expected according to the theory the number of true outliers not detected did not increase as the confidence level decreased. The theory also states that a smaller confidence level increases the chance that true outliers are not detected.
- *Recommended values for the tested parameters.*
  According to the tests the best results are obtained when the number of pings read into the buffer is set at fifty and the test criterion is set at 2.57. As mentioned above there is not a number of neighbours that produces the best result. When taking into account that only a limited number of tests have been performed the usage of a test criterion of 2.57 can be risky. The risk lies in the chance that true outliers that are not detected increases as the value for the test criterion increases. Thus a value of 1.96 for the test criterion is recommended. As mentioned before there is no clear trend for the number of neighbours used and there is not a number of neighbours that generally produce the best results. It is thus not possible, until further research, to recommend a value for the number of neighbours to be used. Fifty pings per buffer is the best number of pings to use based on the performed tests. But the usage of hundred pings will most probably give even better results since the trend is: the more pings used, the les outliers are detected.

- *Comparison between the 2D and 1D algorithm.*
  If the test criterion is kept for both algorithms at 1.96 and for the 2D algorithm six neighbours and fifty pings are used, then the 2D algorithm detects 66 %, 62% and 42% less outliers in data sets 1, 2 and 3 respectively. For all the tests concerning datasets 1 and 2, with a test criterion of 1.96, the improvement was minimally 50 and 40 percent respectively. For data set 4 the 2D algorithm detects more outliers which are situated at the edge of the pipe. Even after further inspection of the results an explanation of these results could not be found and more tests with different data sets are needed to determine if this happens more often and if so with which types of data sets. But on the whole the tests do demonstrate that the 2D outlier detection algorithm is an improvement on the 1D algorithm.
- *Computational time needed for the 2D algorithm.*
  The algorithm was programmed in Matlab. The code was also not optimal in the sense of computing time. The algorithm was thus quite slow as soon as buffer with a thousand or more soundings were used. After optimizing the code for the 2D algorithm the computing time is not expected to be much more than for the 1D algorithm. This is based on the fact that the number of predicted depths calculated is in total the same for both algorithms. The number of calculations needed for the calculation of a covariance function is more for the 2D algorithm since it is based on more soundings. However the number of covariance functions calculated for a data set is more for the 1D algorithm, one for each ping.

7.2 Recommendations

During the development of the algorithm various parameters were revealed. As the amount of time available for the master thesis was limited not all these parameters could be investigated. Therefore in this section some propositions for further research is given.

- *What is the influence of the number of neighbours used?*
  As mentioned in the previous section this is still not clear. The results of the test performed do not show a trend. Since this is actually against the expectation that the more neighbours used the better the results, as is the case with the 1D algorithm, this aspect deserves further investigation. In further tests the number of neighbours used should be increased as well as the number of datasets used. In these test the weights attached to the neighbours should be inspected as well since according to the 1D algorithm the more neighbours that are used the smaller the weights will become of the furthest neighbours and its influence will be negligible.
- *Are the correct mandatory neighbours used?*
  Are enough mandatory neighbours used?
  The idea behind the mandatory neighbours is that the neighbours used must lie in two directions. There is no question about that, but the difficulty is, should the closest neighbour in the next ping be used or the sounding with the same beam number? Also should only four mandatory neighbours be used or rather eight. If eight which eight, the two closest neighbours in the next ping and in the previous ping together with the two closest to the left and right of the sounding in the same ping? Or rather the four closest neighbours in either the next or previous ping together with the four closest in the same ping regardless if it is to the left or right side of the sounding? Again the weights attached to the neighbours should be investigated as well.

- *Are more than 50 pings in a buffer feasible?*
  The tests showed that the more pings used in a buffer, the less outliers were detected. It is thus logical to say that using hundred pings will decrease the number of outliers detected even more. But further tests will have to show if it is feasible in practise to use hundred pings or more since the amount of time needed for the test increases as well
- *Is it possible to use the covariance function of one buffer in the next four buffers?*
  The covariance function often does not differ a lot between buffers and might thus be calculated for one buffer and used in, for example, the following four buffers. In order to answer this question the impact of small changes of the two parameters of the analytical covariance function with respect to the cross validation results should be investigated. A manner in which to do this is to use one covariance function for one data set and alter each parameter separately. The results should then give an indication of the error introduced if the wrong parameter values are used and thus if it is possible to use one covariance function for multiple buffers. Another aspect that might influence the answer is the point noise since it also differs for each buffer.
- *Is the used analytical covariance function the best option?*
  The used analytical covariance function is obtained from the master thesis of P.D. Bottelier. After minor testing and visual inspection of how the function represents the empirical covariance function this function seemed appropriate. During these minor tests a tenth degree polynomial was fitted to the empirical covariance values. This produced a similar representation of the empirical values as the used analytical covariance function. Thus further testing if this produces better results is still open for discussion. A drawback for a polynomial fit can be the computation time.
- *Is it feasible to make the number of beams read into the buffer a parameter?*
  As mentioned in section 7.2 it was decided against a variable number of beams for this project but further study might suggest otherwise. It might be feasible if it is known that a feature (a pipe) is only present in the left side of the data set or if the number of beams in a ping is greater than 100.
- *Is it possible to link the correlation length of the analytical covariance function to the radius used to limit the possible neighbours?*
  In the algorithm the radius is currently a user defined parameter. In the tests it was set at 3 metres. If the distance between the neighbour and the cross validated sounding is greater than the correlation length then the weight attached to the neighbour might be negligible. If this is true then the radius will be adjusted for each buffer separately and the number of possible neighbours decreased. This will in turn reduce the time needed to compute the neighbours.
- *Will a ‘moving’ buffer give better results?*
  Currently cross validation is performed per buffer using all the soundings present in the buffer. Another option is to read sixty pings into the buffer and calculate a covariance function based on the soundings present in those sixty pings. Cross validation will however only be performed on fifty pings, pings six to fifty five. The next buffer will contain the last ten pings of the current buffer and the next forty pings. In this buffer the last five pings of the previous buffer will then be cross validated.

References

H. Bisquay, X. Freulon, C. de Fouquet, C. Lajaunie
Multibeam Data Cleaning for Hydrography using Geostatistics
Proceedings Oceans '98, IEEE, Nice, 1998
http://www.shom.fr/fr_page/fr_act_acquisition/smf/oceans98.htm (September 2003)

P.D. Bottelier
Efficient near real-time processing of high density data with the accent on multibeam
Master thesis at TU Delft, 1998

B.R. Calder, L.A. Mayer
Robust Automatic Multi-beam Bathymetric Processing
International Hydrographic Review, LXIX(2). P 67-76, 1992
http://www.thsoa.org/pdf/h01/3_4.pdf (September 2003)

J.P Chilès, P. Delfiner
Geostatistics: modeling spatial uncertainty
John Wiley & Sons, New York, 1999

N. Debese, H. Bisquay
Automatic detection of erroneous soundings in multibeam data through a robust estimator
Proceedings Hydro-99, Plymouth, UK, 5-7 January 1999

L.L. Dorst
Zeebodem monitoring met geostatistiek en deformatieanalyse
Rapport Koninklijke Marine, Dienst der Hydrografie, 2003

W. Heiskanen, H. Moritz
Physical Geodesy
Technical University Graz, Austria, 1990

D.R. Herlihy, T.N. Stepka, T.D. Rulon
Filtering erroneous soundings from multibeam survey data
Proceedings U.S. Hydrographic Conference '92, 1992

E.H. Isaaks, R.M. Srivastava
An introduction to Applied Geostatistics
Oxford University Press, New York, Oxford, 1989

C.D. de Jong, G. Lachapelle, S. Skone, I.A. Elema
Hydrography
DUP Blue Print, 2002

E.J. de Min
De geoide voor Nederland,
Doctoral thesis at TU Delft, 1996

H. Moritz
Advanced Physical Geodesy
Abacus Press, Tunbridge Wells, Kent, 1980

P.J.G. Teunissen
Adjustment theory; an introduction
Delft University Press, Delft, 2000

P.J.G. Teunissen
Testing theory; an introduction
Delft University Press 2000, Delft

P.J.G. Teunissen
Dynamic data processing; recursive least-squares
Delft University Press 2001, Delft

Tianhang Hou, Lloyd C. Huff, Larry Mayer
Automatic Detection of Outliers in Multibeam Echo Sounding Data
Proceedings U.S. Hydro 2001, Norfolk, VA, USA, May 21-24, 2001

Software:

Matlab
Version 6.1.0.450
Release 12.1

Terramodel
Version 9.70
Toolpak version 4.40

Surface
Version 2.12.137
Module of Starfix Suite

Appendix A

In this appendix 8 Digital Terrain Models (DTM's) are being displayed graphically showing the results of 8 tests. There are 2 DTM's per data set, one describing the reference test (1D algorithm) and one describing the 2D algorithm. For the tests concerning the 2D algorithm the configuration settings were equal and set at:

- The number of pings: 50
- The number of neighbours: 6
- The test criterion value: 1.96.

Comment

In the DTM of the second data set after data cleaning with the 2D algorithm, Figure A4, large triangles are visible at the left and right side of the DTM. These large triangles are the results of outliers still present in the data set. The DTM of the reference results (Figure A3) does not have these large triangles due to the fact that the first and last two soundings are not tested and not displayed either.

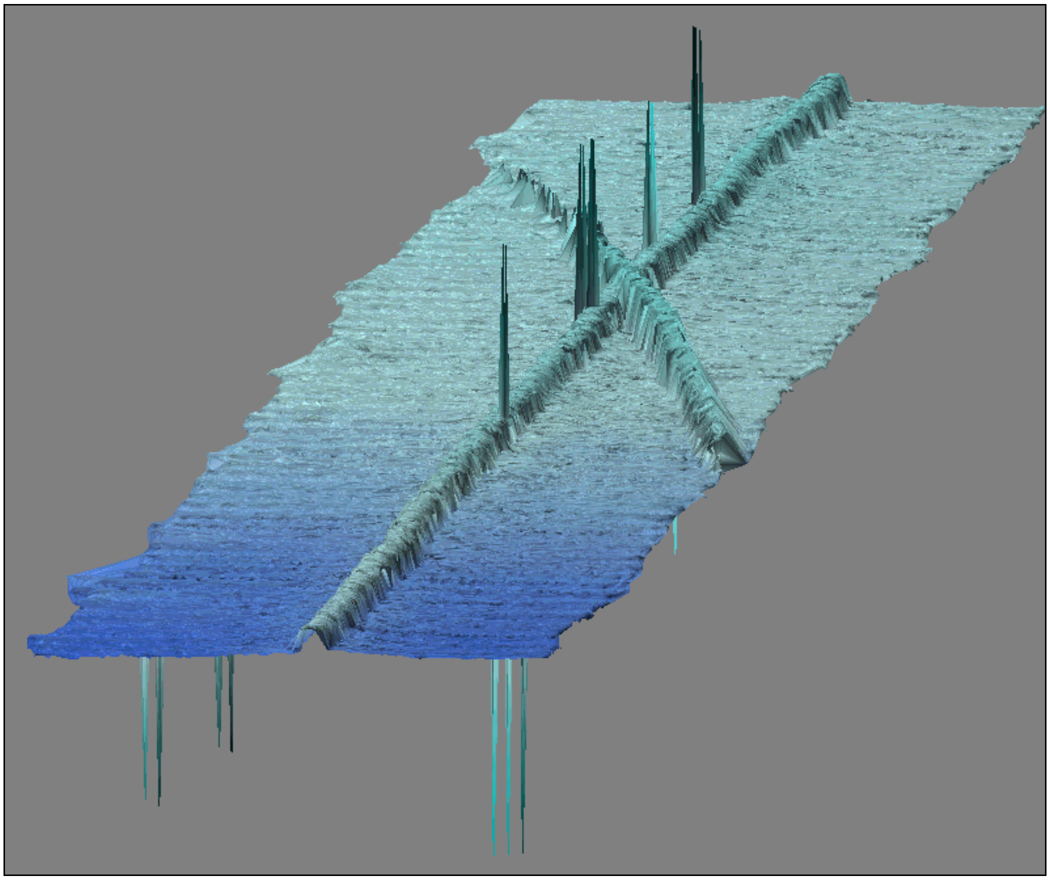

Figure A1 DTM of data set 1 after data cleaning using the 1D algorithm (the reference DTM)

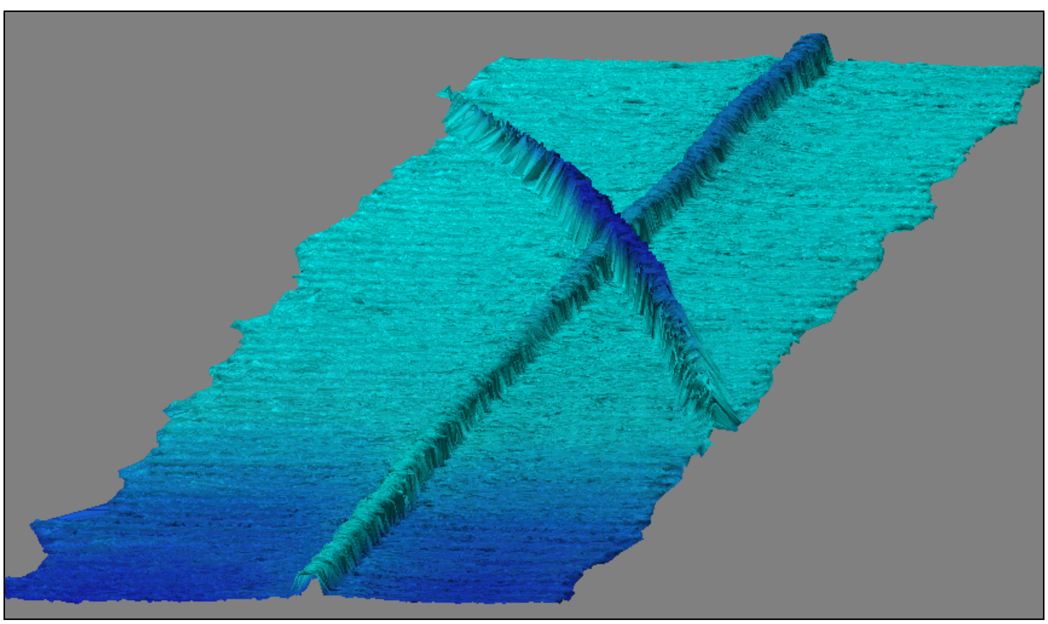

Figure A2 DTM of data set after data cleaning using the 2D algorithm

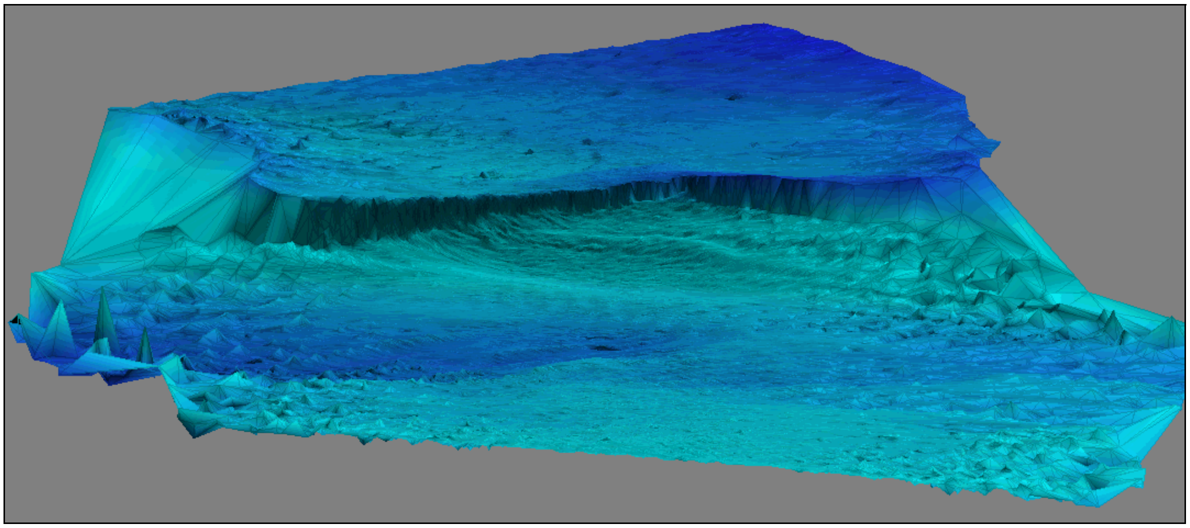

Figure A3 DTM of data set 2 after data cleaning using the 1D algorithm (the reference DTM)

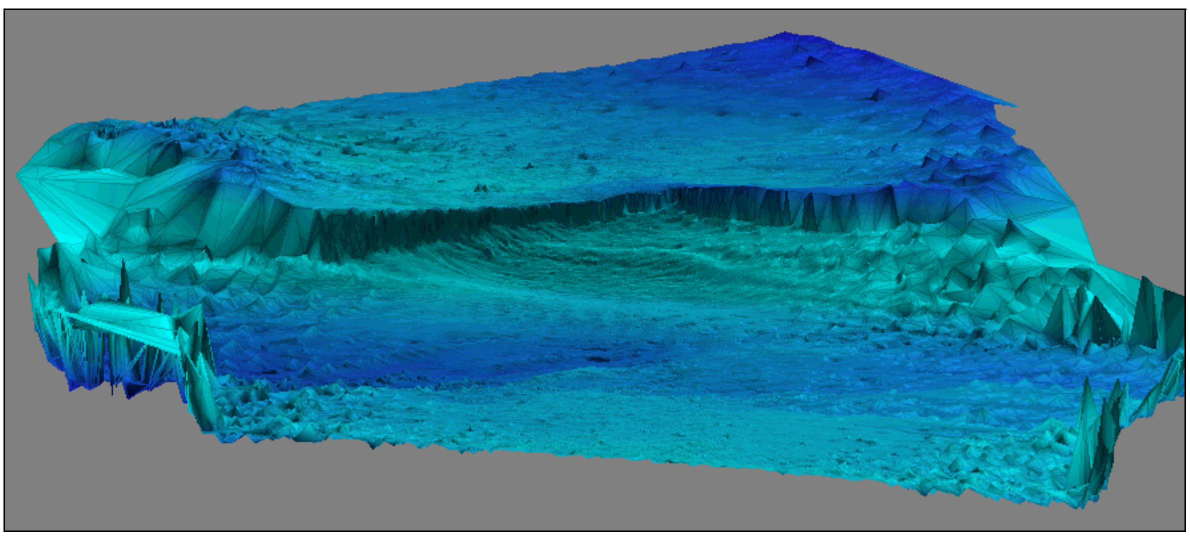

Figure A4 DTM of data set 2 after data cleaning using the 2D algorithm

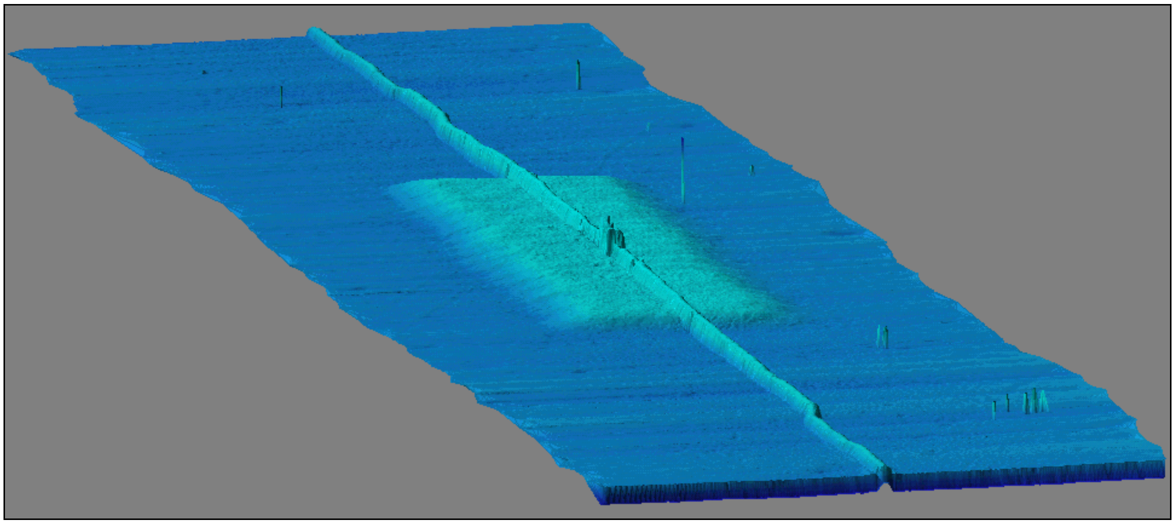

Figure A5 DTM of data set 3 after data cleaning using the 1D algorithm (the reference DTM)

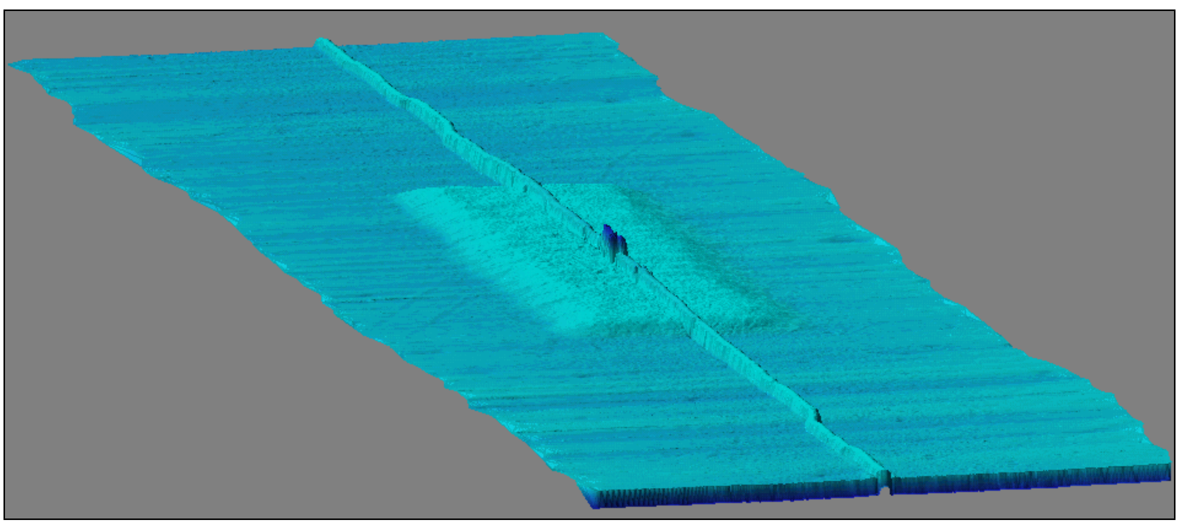

Figure A6 DTM of data set 3 after data cleaning using the 2D algorithm

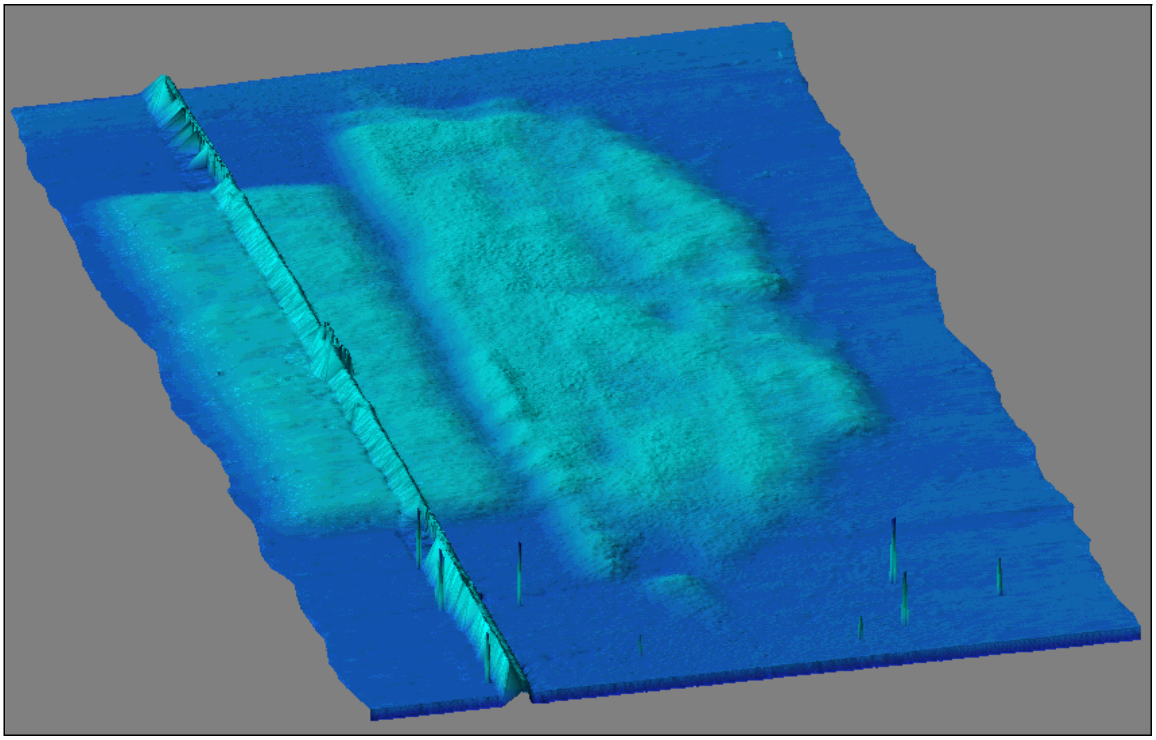

Figure A7 DTM of data set 4 after data cleaning using the 1D algorithm (the reference DTM)

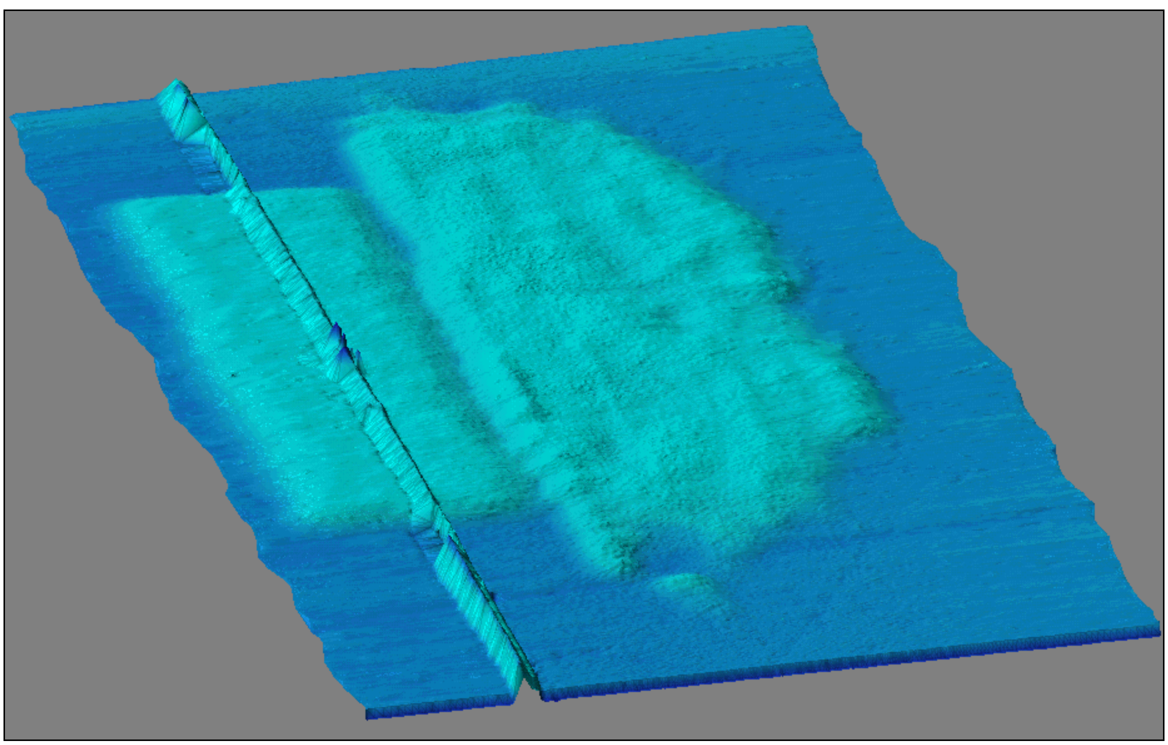

Figure A8 DTM of data set 4 after data cleaning using the 2D algorithm

Appendix B

In this appendix 8 top views will be given of the results of 8 tests. In these top views the green dots are the soundings which are accepted and the red dots the soundings that are detected as outliers. There are 2 top views per dataset, one of the reference test (1D algorithm) and one of the 2D algorithm. For the tests concerning the 2D algorithm the configuration settings were equal and set at:

- The number of pings: 50
- The number of neighbours: 6
- The test criterion value: 1.96

Comment

It is possible to see that the outliers detected lie mainly at the edges of the pipes. It can also be seen that after data cleaning with the 2D algorithm the number of outliers detected along the edges are less then when the 1D algorithm is used. In data set 2 the 1D algorithm also detects outliers in the middle of the data set (Figure B3) whereas the 2D algorithm does not detect them (Figure B4).

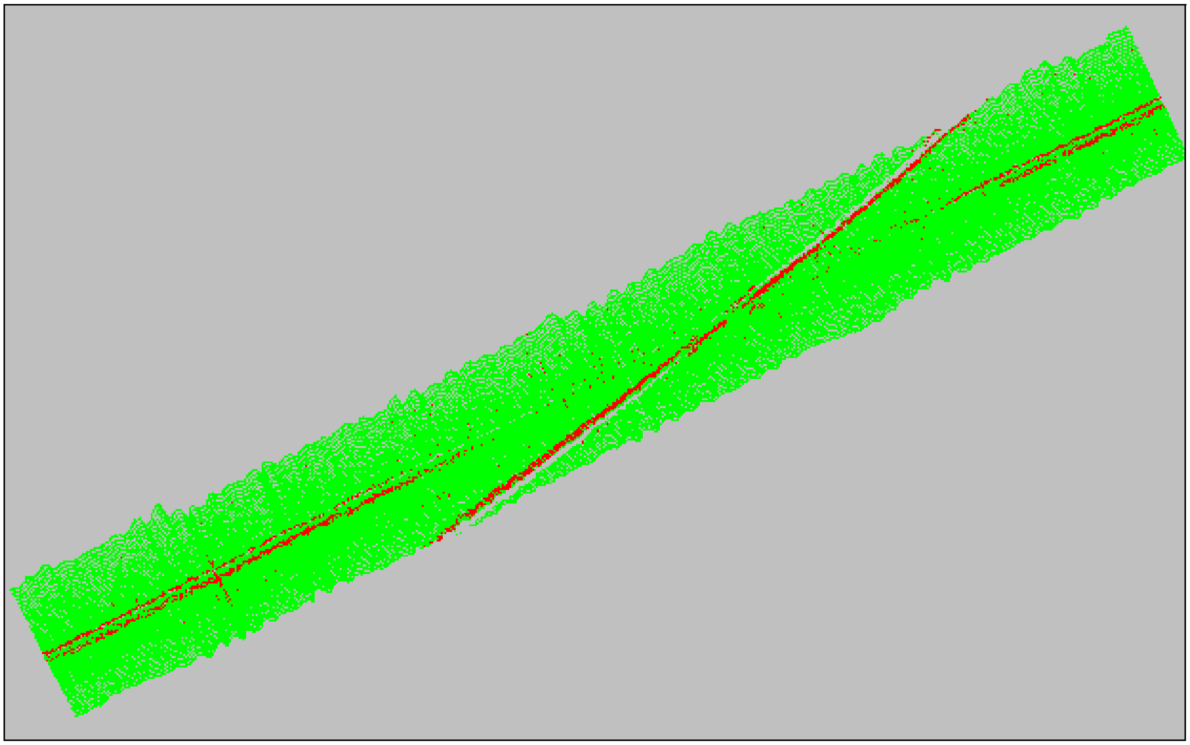

Figure B1 Top view of data set 1 after data cleaning using the 1D algorithm (the reference DTM)

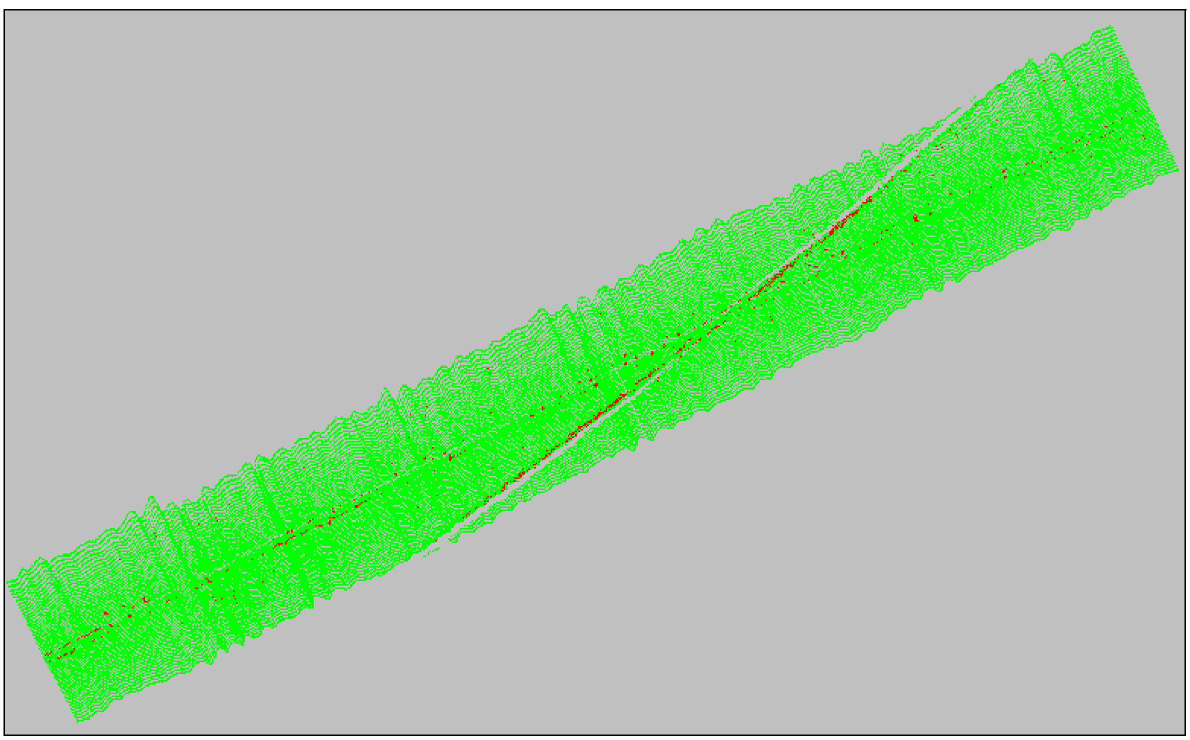

Figure B2 Top view of data set 1 after data cleaning using the 2D algorithm

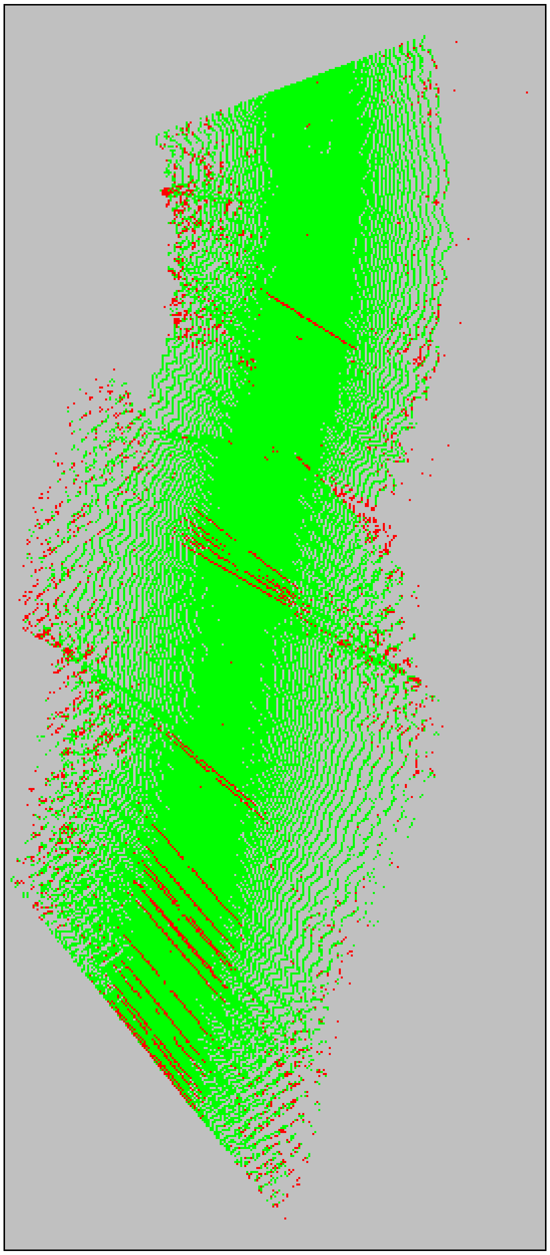

Figure B3 Top view of data set 2 after data cleaning using the 1D algorithm (the reference DTM)

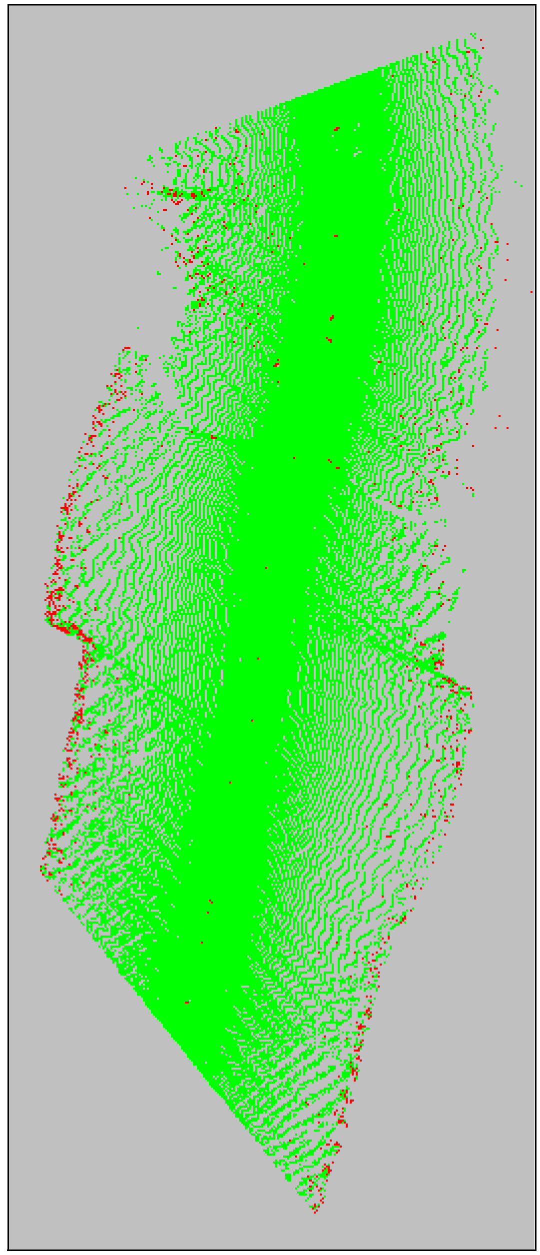

Figure B4 Top view of data set 2 after data cleaning using the 2D algorithm

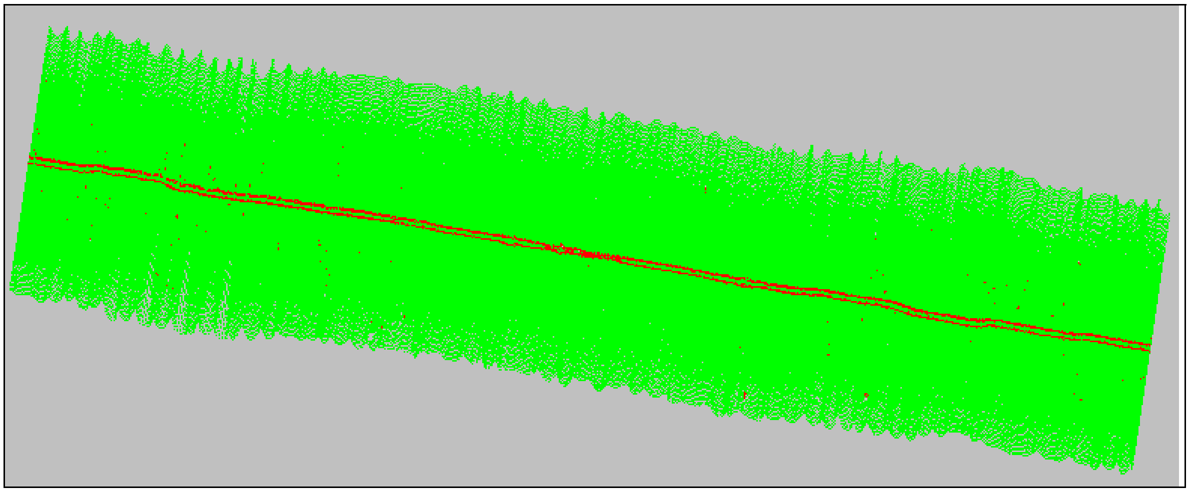

Figure B5 Top view of data set 3 after data cleaning using the 1D algorithm (the reference DTM)

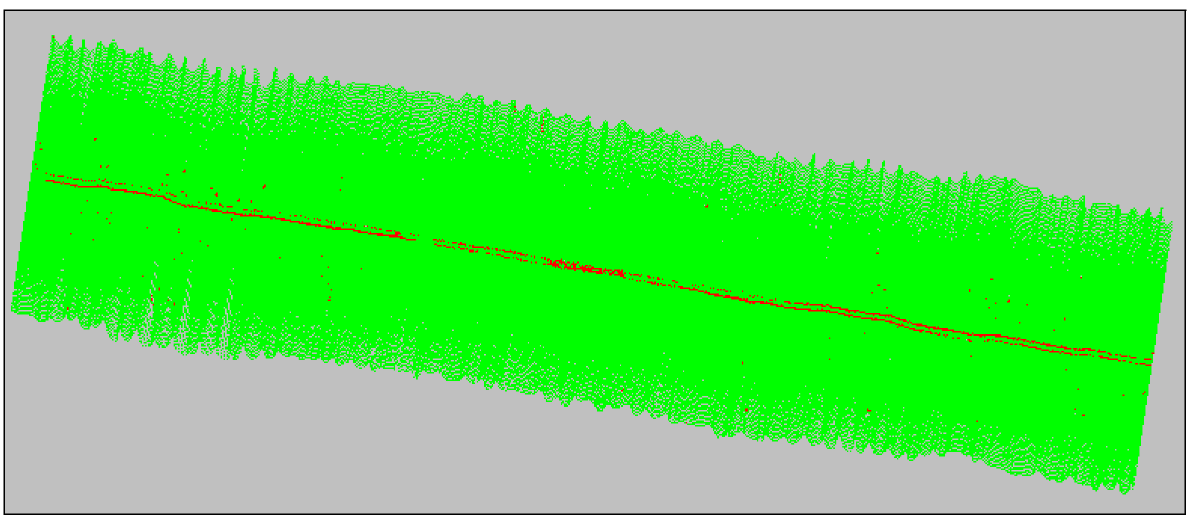

Figure B6 Top view of data set 3 after data cleaning using the 2D algorithm

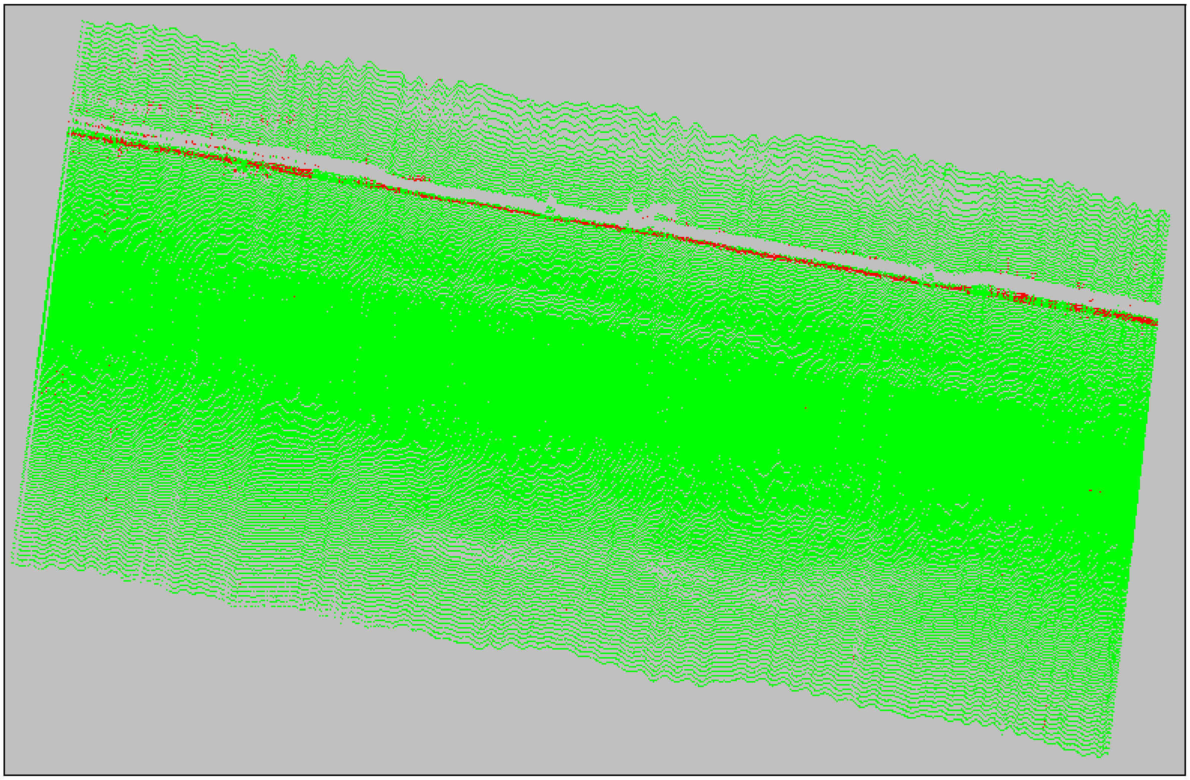

Figure B7 Top view of data set 4 after data cleaning using the 1D algorithm (the reference DTM)

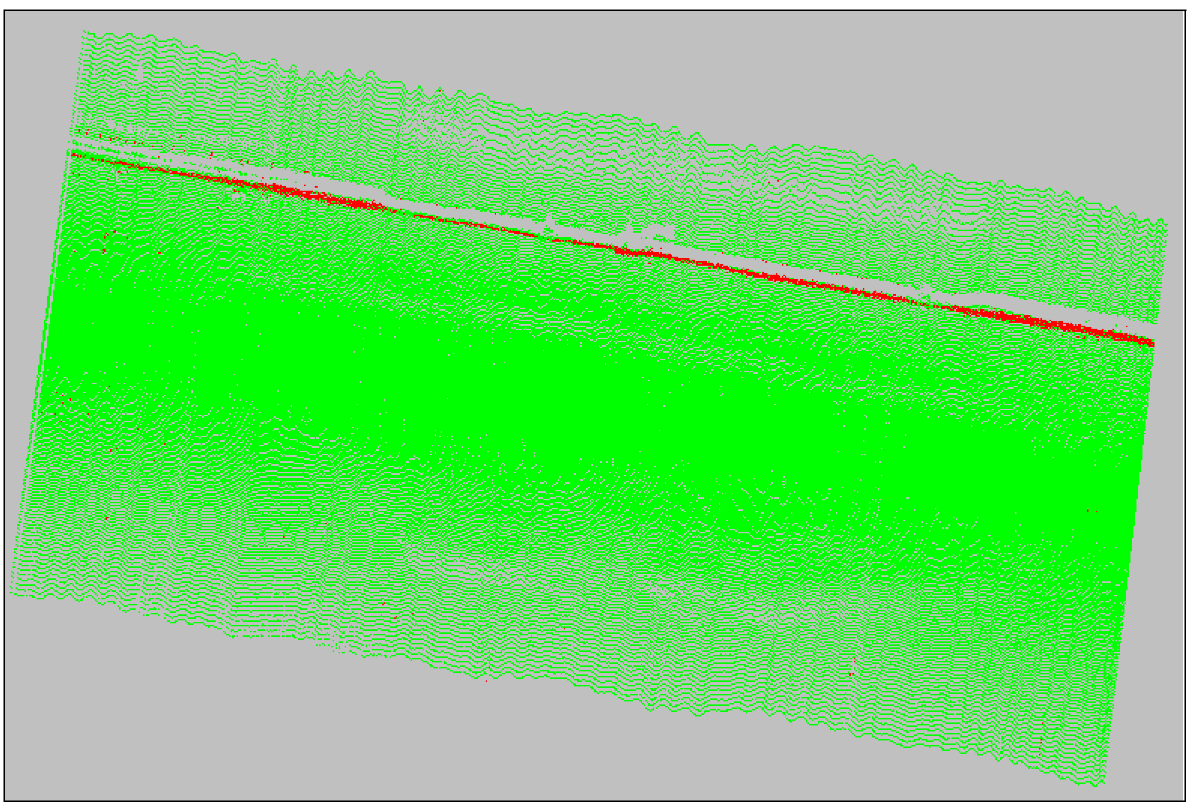

Figure B8 Top view of data set 4 after data cleaning using the 2D algorithm